KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Tensorization

and applications in blind source separation

Otto Debals

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

August 2017

Tensorization

and applications in blind source separation

Otto DEBALS

Examination committee:
Prof. dr. ir. J. Berlamont, chair
Prof. dr. ir. L. De Lathauwer, supervisor
Prof. dr. ir. M. Van Barel, co-supervisor
Prof. dr. ir. M. Moonen
Prof. dr. ir. S. Van Huffel
Prof. dr. S. Theodoridis
(National and Kapodistrian University of Athens, Greece)
Prof. dr. A. Belouchrani
(Ecole Nationale Polytechnique, Algeria)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

August 2017

© 2017 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Otto Debals, Kasteelpark Arenberg 10 box 2440, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Preface

The foundations of this thesis were established in my hometown Kortrijk, Belgium. I had the pleasure of having Dr. Fabien Decruyenaere as mathematics teacher in secundary school. He inspired me to achieve my full potential, and at university, I worked hard to get good results. During those first bachelor years, I was especially attracted by the courses from Prof. Lieven De Lathauwer. I loved those courses. Clear, crisp mathematics. Advanced, yet not too abstract and overcomplicated. I rediscovered the verge between mathematics and signal processing in the master thesis which I wrote with Nico, under the guidance of Lieven. Together with the pleasant commitment towards Sagio.be, that final master's year cannot be described as relaxing and peaceful at all. Things worked out eventually, especially after meeting Astrid that year.

I decided to postpone my academic farewell and I started a PhD in Lieven's group in September 2013 with Prof. Marc Van Barel as co-supervisor. It kicked off on a sad note by not being awarded a grant from the Research Foundations – Flanders (FWO). My disappointment at the time was a public secret. But by failing forward, I did manage in December that year to have my PhD personally funded by the Agency for Innovation and Entrepreneurship, formerly known as the Agency for Innovation through Science and Technology (IWT). I am very grateful for this opportunity, allowing me to independently perform research while having a number of goals and deliverables at hand. I have to especially thank Devy, Kirsten, Ninah, Steven and the others for taking the time to prepare me for the IWT defense.

These four years of PhD were both challenging and interesting. It must be said that it took me a while to decide to go for a PhD, but I can now say it was totally the right choice at the right time. In the remainder of this preface, I would like to thank the people that have made my PhD worth it.

First, Lieven, thank you for allowing me to join your young and small but energizing research group. Thanks for the informal discussions and the necessary patience you had. Your perfectionism was challenging to pursue yet easy to admire. Marc, you deal with a great combination of calmness, enthusiasm and positiveness. I enjoyed the meetings with you every single bit. Prof. Sabine Van Huffel, thanks for joining my supervisory committee and a special thank-you is appropriate for allowing us to join your group's social events. Sabine, your smile and laugh are heartwarming. Prof. Marc Moonen, thanks for being part

of my committee as well. Your thorough and compelling feedback during the sporadic encounters is often challenging but never misplaced. I would especially like to thank the external members of the committee, Prof. Sergios Theodoridis and Prof. Adel Belouchrani, for joining the defense(s) in Belgium, as well as the chair Prof. Jean Berlamont. Furthermore, thanks Marleen, Ida, Elsy, John, Wim, Maarten, Jacqueline and other back-office people for the spurious efforts. It certainly made my practical and administrative issues less a burden.

The PhD has allowed me to travel to different locations. Hong Kong, pearl of the orient, blew me away. It was my first encounter with top-notch researchers in the field. Brisbane (Australia) was host of my first signal processing conference. Thanks Wouter, Amin, Enzo and Jorge for accompanying me. The Dolomites (Italy) formed a closer yet most appealing location for the 2015 TRICAP conference, the latter consisting of a select group of tensor-focused researchers. I remember a chat with Prof. Rasmus Bro regarding food science, discussing the connections between tensors and edible fermented moss during a hike. Other conferences and summer schools followed in Prague (Czech Republic), Göttingen and Bonn (Germany; thanks Ben, Daan and Martijn!), Lommel and Leuven. I am especially indebted to Prof. Eleftherios Kofidis for allowing me to enjoy a research trip to Athens. Thanks Christos and Manuel for having lunch together; Christos, I still owe you some Belgian beers! I enjoyed the daily chats when you passed by for coffee, Sergios; your dedication towards 'your' engineering department should be honored. Leftheris, I will cherish the amazing dinner we had in the local $\psi\alpha\rho\sigma\tau\alpha\beta\acute{\varepsilon}\rho\nu\alpha$ (fish tavern). Finally, we have to admit that one can finish worse with a conference on the beautiful island group of Hawai'i (USA); thanks Bert, Davy, Rahaf, Ali and Vivek for the joyful evenings!

Our designated working places within the department in Leuven were not less diverse as the locations abroad. After graduation, Nico and I spent a couple of weeks working in a small meeting room. Thanks Nico for (un)willingly occupying the smaller of both tables. Soon, we moved to an office with Yunlong Feng and Yuning Yang. Thanks guys for the amazing dinners we had; the chicken feet marinated in Coca Cola were interesting, to say the least. Emanuele Frandi and Paul Smyth joined our office later, the latter also joining our research team. Thanks for bringing in the life experience, Paul. When the new part of ESAT was ready, we were again moving ... to an even older part of the building. However, it felt good at the fifth floor in the tower and everyone was at ease quite quickly. We got used to the stairs, except when carrying water bottles.

I had the chance to combine my PhD with some extra-PhD items. Thanks Bart, for introducing me to CFA; it definitely brought me insightful knowledge on financial markets. Laurent, Bert and Marc: thanks for the numerous meetings we had, trying to launch Investimize. We did not succeed, but we can agree that we all learned a lot. Who knows the journey might continue in the future? David,

I'm glad we met during an Lcie event, resulting in founding TechStart together with Tom and Steven; thanks Wim for the support and Christof, Joachim, Caspar, Laurent and Mattia for continuing our legacy. Thanks Jonathan, Kin Chi, Tom, Filip, Marc and others in the board of directors for joining me in the quest to let Sagio.be flourish. Friends from the water-polo, thanks for providing me with the weekly opportunity of practicing my favorite sport. Lucas, Janis and Astrid, we can be proud of ourselves for finishing that enviable marathon of Athens. It's pretty certain: that was a damn long hill. Thanks all of you yuppies (Sven, Ce, Bert, Stals, Jennes, Carl, Palmen) for the nights out and the (not-so-)mystery weekends — where will the next mystery weekend take us? — and the KULAK burgies for the various barbecues and re-energizing weekends. Stein, let's continue the runs, drinks, dinners and sporadic encounters.

A number of constants dominated the previous four years. Of course, Nico, thanks for playing the silent monkey more often than you wanted — read: being silent and just sitting next to me while I'm explaining something out loud, allowing me to create my own insights. I hope you learned something as well ;-). Frederik and Martijn, I enjoyed the first year as your master thesis supervisor and the other years as your colleague. Nico, Fre, Martijn, please all follow your passion in the rest of your careers, whether it is code optimization, web development or graphic design, respectively. One guy quickly understood that Nespresso was his favorite coffee brand, resulting in a daily visit to our office: Tom. Thanks for the coffee breaks, and thanks for often disturbing me when focusing on work. Your friendship means a lot to me. A special thanks goes to Griet for organizing the various social events within the Biomed group (Thanks Rob, Alex, Carolina, Bori, Laure, Thomas, Bharath, those were already there and those who joined recently!). Colleagues from KULAK (Ignat and Mikael, later on also Xiaofeng, Michiel, Chuan and Alwin), we might have spoken more through mail than face-to-face but I enjoyed our encounters every single time, whether it was you coming to Leuven or me traveling to Kortrijk.

Finally, a special thanks goes to my family and family-in-law to-be. Thank you mama and papa. I dedicate this manuscript to you. You have allowed me to become the person I am now, and you have laid out the foundations for my academic career. You were there whenever needed. Benno, we differ in age quite a bit, but our humor and passions are very alike and we definitely share more characteristics than one might initially think of. Remember, just do what you love, and try to do it well. And of course, Astrid, love of my life. Thanks, for being there. Thanks for your love, patience, trust and joy. I'm proud to become your lawfully wedded husband.

Thanks everyone. Mahalo and aloha,

Otto

Abstract

The value of data cannot be underestimated in our current digital age. Data mining techniques have allowed various priceless technological advances, influencing our daily lives to a significant extent. An important aspect of data mining is data representation. While vectors and matrices can be used to represent one-way and two-way data, respectively, so-called tensors are well suited to represent multiway data. The capabilities of recently developed tensor tools such as tensor decompositions surpass the power of their vector and matrix counterparts. It follows that these tools are already established in domains such as signal processing, statistics and machine learning.

Tensor tools obviously require a tensor. In various applications such as source separation and data clustering, only one- or two-way data is available. While classic matrix tools sometimes fall short, tensor tools have the ability to, for example, uniquely identify underlying components. The main goal of this thesis consists of investigating how one can purposefully use tensor tools and exploit their powerful tensor properties in such applications given only a single vector or matrix. One approach encompasses a so-called tensorization step by first mapping the given data to a tensor. A number of tensorization techniques have appeared in the literature such as Hankelization and higher-order statistics. Not every mapping is meaningful though, and the effectiveness of a technique strongly depends on the problem at hand.

In this thesis we present a comprehensive overview of both existing and novel tensorization techniques. We uncover relations between the properties of the given data and the properties of the tensor obtained after tensorization, and provide connections with tensor tools. We showcase the power of tensorization in the context of instantaneous and convolutive blind signal separation, including fetal heart rate extraction, direction-of-arrival estimation and blind separation of 16-QAM signals, and provide theoretical working conditions. Other applications are touched upon as well, such as data and graph clustering and the training of neural networks. Furthermore, we exploit our expertise in tensor-based optimization to propose a novel technique for nonnegative matrix factorization. Throughout the thesis, particular attention is paid to the use of tensorization in a large-scale context, leading to efficient representations of structured tensors and algorithms that are able to cope with large tensors after tensorization.

Beknopte samenvatting

De waarde van data kan niet onderschat worden in het huidige digitale tijdperk. Technieken voor dataontginning hebben geleid tot verschillende waardevolle technologische vooruitgangen die ons dagelijks leven significant hebben beïnvloed. Een belangrijk aspect van dataontginning is datarepresentatie. Terwijl vectoren en matrices gebruikt kunnen worden om respectievelijk één- en tweewegsdata te beschrijven, zijn zogenoemde tensoren uitermate geschikt om meerwegsdata voor te stellen. De mogelijkheden van recent ontwikkelde tensorinstrumenten zoals tensorontbindingen overtreffen de eigenschappen van hun vector- en matrixequivalenten. Deze instrumenten zijn daarom reeds gevestigde waarden geworden in domeinen zoals signaalverwerking, statistiek en *machine learning*.

Tensorinstrumenten vereisen vanzelfsprekend een tensor. In verscheidene toepassingen zoals signaalscheiding en dataclustering is enkel één- of tweewegsdata beschikbaar. Terwijl klassieke matrixtechnieken soms tekortschieten, hebben tensorinstrumenten de kracht om, bijvoorbeeld, onderliggende componenten op een eenduidige manier te identificeren. Het hoofddoel van deze thesis bestaat in het onderzoeken hoe, gegeven een enkele vector of matrix, tensorinstrumenten en hun krachtige tensoreigenschappen gebruikt kunnen worden. Een mogelijke aanpak omvat een zogenoemde tensorisatiestap door eerst de gegeven data om te vormen naar een tensor. Een aantal tensorisatietechnieken zijn reeds verschenen in de literatuur zoals Hankelisatie en hogere-ordestatistieken. Niet iedere omvorming is echter betekenisvol en de doeltreffendheid van een techniek hangt sterk af van het beschouwde probleem.

In deze thesis stellen we een uitgebreid overzicht voor van bestaande en nieuwe tensorisatietechnieken. We leggen relaties bloot tussen eigenschappen van de gegeven data en die van de tensor na tensorisatie, en leggen verbanden met tensorinstrumenten. We demonstreren de kracht van tensorisatie aan de hand van ogenblikkelijke en convolutieve blinde signaalscheiding, inclusief hartsignaalscheiding, invalshoekschatting en 16-QAM scheiding, en voorzien theoretische werkingsvoorwaarden. Er worden ook kort andere toepassingen aangeraakt zoals data- en grafenclustering en neurale netwerken. Bovendien buiten we onze expertise in tensorgebaseerde optimalisatie uit om een nieuwe techniek voor te stellen voor niet-negatieve matrixfactorisatie. Doorheen de thesis wordt aandacht besteed aan het gebruik van tensorisatie voor grootschalige data. Dit leidt tot efficiënte voorstellingen van gestructureerde tensoren en algoritmen die vlot grote tensoren na tensorisatie kunnen verwerken.

List of Abbreviations

A	**ACMA**	Analytical constant modulus algorithm
	aHALS	Accelerated hierarchical alternating least squares
	ALS	Alternating least squares
B	**BCA**	Block component analysis
	BFGS	Broyden–Fletcher–Goldfarb–Shanno
	BPSK	Binary phase-shift keying
	BSS	Blind source separation
	BSI	Blind system identification
	BTD	Block term decomposition
C	**CCA**	Canonical correlation analysis
	CCPD	Coupled canonical polyadic decomposition
	CDMA	Code division multiple access
	CFSK	Continuous frequency-shift keying
	CG	Conjugate gradients
	CM	Constant modulus
	CMA	Constant modulus algorithm
	CMTF	Coupled matrix and tensor factorization
	CPD	Canonical polyadic decomposition
	CPSK	Continuous phase-shift keying
	CWT	Continuous wavelet transform

D	**DFT**	Discrete Fourier transform
	DOA	Direction of arrival
	DS-CDMA	Direct sequence CDMA
	DWT	Discrete wavelet transform
E	**ECG**	Electrocardiography
	ECoG	Electrocorticography
	EEG	Electroencephalography
	EEM	Excitation–emission matrix
	EMD	Empiral-mode decomposition
	EVD	Eigenvalue decomposition
F	**FECG**	Fetal electrocardiography
G	**GEVV**	Generalized eigenvalue decomposition
	GN	Gauss–Newton
H	**HOS**	Higher-order statistics
I	**ICA**	Independent component analysis
	IMF	Intrinsic mode function
J	**JADE**	Joint approximation diagonalization of eigenmatrices
L	**LM**	Levenberg–Marquardt
	LMLRA	Low multilinear rank approximation
	LS	Line search or least squares

M	**MHR**	Multidimensional harmonic retrieval
	MIMO	Multiple-input multiple-output
	MLSVD	Multilinear singular value decomposition
	MM	Multi-modulus
	MMA	Multi-modulus algorithm
	MRI	Magnetic resonance imaging
	MU	Multiplicative updates
N	**NCG**	Nonlinear conjugate gradient
	NLS	Nonlinear least squares
	NMF	Nonnegative matrix factorization
	NP-NMF	Nonnegative polynomial-based NMF
P	**PCA**	Principal component analysis
	PD	Polyadic decomposition
	PG	Projected gradients
	PS	Plane search
Q	**QAM**	Quadrature amplitude modulation
	QN	Quasi-Newton
	QPSK	Quadrate phase-shift keying
	qTT	Quantized tensor train
R	**RMSE**	Root mean square error
S	**SCA**	Sparse component analysis
	SDF	Structured data fusion
	SISO	Single-input single-output

S	**SNR**	Signal-to-noise ratio
	SOBI	Second-order blind identification
	SOS	Sum of squares
	STFT	Short-time Fourier transform
	SVD	Singular value decomposition
	SVM	Support vector machines
T	**TT**	Tensor train
U	**ULA**	Uniform linear array
	URA	Uniform rectangular array
W	**WBAN**	Wireless body area network
	WVD	Wigner–Ville distribution

List of Symbols

Object notation

$A, B, \ldots$	Integers
$a, b, \ldots$	Scalars
$\mathbf{a}, \mathbf{b}, \ldots$	Column vectors
$\mathbf{A}, \mathbf{B}, \ldots$	Matrices
$\mathcal{A}, \mathcal{B}, \ldots$	Higher-order tensors
$\underline{f}$, $\underline{\mathbf{f}}$, $\underline{\mathcal{F}}$	Scalar-, vector- and tensor-valued function
$\mathbf{I}_N$	Identity matrix of size $N \times N$
$\mathbf{1}_N$	$N \times 1$ column vector with all ones
$\mathbf{0}_N$	$N \times 1$ column vector with all zeros
$\mathbf{1}_{M \times N}$	$M \times N$ matrix with all ones
$\mathbf{0}_{M \times N}$	$M \times N$ matrix with all zeros
$\mathbf{e}_i$	ith canonical basis vector

Notation of operators with one variable

$\lvert\cdot\rvert$	Absolute value
$\lvert\lvert\cdot\rvert\rvert$	Frobenius norm
$\overline{\cdot}$ and $\cdot^\star$	Complex conjugate

$.^{\mathrm{T}}$	Transpose
$.^{\mathrm{H}}$	Hermitian conjugate
$.^{-1}$	Matrix inverse
$.^{-\mathrm{T}}$	Transposed inverse
$.^{\dagger}$	Moore–Penrose pseudoinverse
$.^{(n)}$	The nth element of a sequence
$\cdot_{(n)}$	Mode-n tensor matricization
$\Re\{\cdot\}$	Real part
$\Im\{\cdot\}$	Imaginary part
$\mathrm{E}\{\cdot\}$	Expected value
$\hat{\cdot}$	Estimated value
$\mathrm{vec}\,(\cdot)$	Column-wise vectorization
$\det(\cdot)$	Matrix determinant
$r(\cdot)$	Matrix rank
$\mathbf{C}_D(\cdot)$	Dth compound matrix
$\underline{f}'(\cdot)$	Total derivative

Notation of operators with multiple variables

$\odot$	Column-wise Khatri–Rao product
$\odot^{\mathrm{T}}$	Row-wise Khatri–Rao product
$\otimes$	Kronecker product
$\circledast$	Outer product
$*$	Hadamard product

$\cdot_n$	Mode-n tensor-matrix product
$\langle \cdot, \cdot \rangle$	Inner product
$\operatorname{diag}(\mathbf{A}, \mathbf{B}, \ldots)$	Block-diagonal matrix with blocks $\mathbf{A}$,$\mathbf{B}$,...
$[\mathbf{A}, \mathbf{B}, \ldots]$	Horizontal concatenation of two or more matrices
$[\mathbf{A}; \mathbf{B}; \ldots]$	Vertical concatenation of two or more matrices
$[\![\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)}]\!]$	CPD with factor matrices $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)}$

Other

$\mathbb{R}$	Set of real scalars
$\mathbb{R}^{I_1 \times \cdots \times I_N}$	Set of real-valued tensors of size $I_1 \times \cdots \times I_N$
$\mathbb{C}$	Set of complex scalars
$\mathbb{C}^{I_1 \times \cdots \times I_N}$	Set of complex-valued tensors of size $I_1 \times \cdots \times I_N$
$\mathbb{K}$	Either $\mathbb{R}$ or $\mathbb{C}$
$\mathcal{O}(\cdot)$	Big O
$:=$	Defined as
$\equiv$	Equivalent with
$\binom{N}{k}$	Binomial coefficient

Contents

Preface i

Abstract v

List of Abbreviations ix

List of Symbols xiii

Contents xvii

List of Figures xxiii

List of Tables xxvii

1 Introduction 1

1.1 The new oil 1

1.2 One-way and two-way data 2

1.3 Multiway data 6

1.4 Crossing the chasm: Tensorization 10

1.5 Research aims 12

1.6 Outline of the thesis 14

2 The concept of tensorization 19

2.1 Introduction 20

2.2 Preliminaries on (multi)linear algebra and multiway analysis . 22

2.2.1 Basic notation and matrix/tensor operations 22
2.2.2 Preliminaries on functions 23
2.2.3 Matrix factorization 24
2.2.4 (Coupled) Tensor decompositions 25
2.3 Multiway experiment design 29
2.4 Functions and derivatives 31
2.4.1 Evaluation of scalar multivariate functions 31
2.4.2 Evaluation of vector-, matrix- and tensor-valued multivariate functions 33
2.4.3 Tensorization using derivatives 34
2.5 Tensors representing mathematical objects 36
2.5.1 Relations between multilinear functions and tensors . . 37
2.5.2 Relations between polynomials and tensors 39
2.6 Tensorization of a single vector 44
2.6.1 Hankel/Toeplitz matrices and tensors 45
2.6.2 Segmentation, decimation, folding and reshaping 50
2.6.3 Löwner matrices and tensors 52
2.6.4 Determinant-defining matrix and monomial relations . . 57
2.6.5 Time–frequency and time–scale techniques 60
2.6.6 Other tensorization techniques 65
2.7 Tensorization of a given set of vectors 66
2.7.1 Collection of matrices obtained by parameter variation . 67
2.7.2 Collection of matrices indirectly resulting in a CPD . . 73
2.7.3 Direct construction of a tensor 78
2.8 Further discussions and future work 89
2.9 Conclusion 90

3 Löwner-based blind signal separation of rational functions 93

3.1 Introduction 94
3.1.1 Notation and basic operations 95
3.1.2 Rank definitions and basic tensor decompositions 96
3.2 Löwner-based blind signal separation 97
3.2.1 The blind signal separation problem 98
3.2.2 Löwner matrices 99
3.2.3 Tensorization and block term decomposition 100
3.2.4 Recovery of the mixing matrix and the sources 100
3.3 Factorization of Löwner matrices 101
3.3.1 Case of rational functions with non-coinciding poles . . 102
3.3.2 General case of rational functions with coinciding poles 103
3.3.3 Case of polynomials 104
3.3.4 Summary and implication for blind source separation . 105
3.4 Uniqueness 106
3.5 Connection with Hankel-based tensorization and Vandermonde decomposition 107
3.5.1 Transformation between Löwner and Hankel 108
3.5.2 Choice of sampling points 108
3.6 Experiments and applications 109
3.6.1 General experiment 110
3.6.2 Underdetermined system 111
3.6.3 Fluorescence spectroscopy 112
3.6.4 Fetal electrocardiogram extraction 114
3.7 Conclusion 116

4 A tensor-based method for large-scale blind source separation using segmentation 117

4.1 Introduction 118

4.1.1 Notation and definitions 119
4.1.2 Tensor decompositions 120
4.2 Large-scale blind source separation via low-rank sources 122
4.2.1 Blind source separation 123
4.2.2 Low-rank sources 123
4.2.3 Decomposition 128
4.3 Large-scale blind source separation via low-rank mixing vectors 129
4.4 Large-scale blind source separation using twofold segmentation 131
4.5 Simulations and applications 132
4.5.1 General experiments 133
4.5.2 Underdetermined mixture 134
4.5.3 Noise and sample length 135
4.5.4 Low-rank approximation 136
4.5.5 Compression versus accuracy 138
4.5.6 Fetal electrocardiogram extraction 139
4.5.7 Direction of arrival estimation 141
4.6 Conclusion 143

5 Analytical multi-modulus algorithms based on coupled canonical polyadic decompositions 145

5.1 Introduction 146
5.2 Multilinear algebra and notation 148
5.3 Tensor-based multi-modulus blind signal separation 149
5.3.1 Translation into a constrained set of linear equations . . 150
5.3.2 Omitting identical columns from the linear system . . . 152
5.3.3 Solving the system 152
5.3.4 Recovery of the separation matrix 153
5.3.5 Interpretation in a tensor framework 154

5.3.6 Note concerning rank deficiency of the mixing matrix . 155
5.4 Multi-modulus BSS using a rank-1 detection procedure 156
5.4.1 Rank-1 detection procedure 157
5.4.2 Application in the multi-modulus setting 159
5.5 Blind deconvolution of multi-modulus signals 161
5.5.1 Data model 161
5.5.2 Deconvolution using an increased number of source signals 162
5.5.3 Deconvolution using the original number of source signals by exploiting the block-Toeplitz structure 162
5.6 Simulations 163
5.6.1 Simulation for rank-1 detection algorithm 164
5.6.2 Simulations for the instantaneous case 165
5.6.3 Simulations for the convolutive case 166
5.7 Conclusion 169

6 Tensorlab 3.0 — Numerical optimization strategies 173

6.1 Introduction 174
6.1.1 History and philosophy 174
6.1.2 Notation 176
6.2 Structured data fusion 177
6.3 Tensorization 178
6.4 Coupled matrix/tensor factorization 180
6.5 Large-scale tensor decompositions 181
6.5.1 Randomized compression 182
6.5.2 Incomplete tensors 183
6.5.3 Randomized block sampling 184
6.5.4 Efficient representation of structured tensors 185
6.6 Conclusion 185

7 Nonnegative matrix factorization using nonnegative polynomial approximations 187

7.1 Introduction 188

7.2 Nonnegative polynomial-based NMF 189

7.2.1 Modeling nonnegative polynomials on (in)finite intervals 189

7.2.2 Connection with discrete-time signals 190

7.2.3 NMF using nonnegative polynomials 190

7.3 Algorithms and computational aspects 191

7.3.1 Quasi-Newton and nonlinear least squares algorithms . 191

7.3.2 Preprocessing with orthogonal compression 193

7.4 Simulations and applications 193

7.5 Conclusion 197

8 Conclusion 199

8.1 Contributions 199

8.2 Prospective work 205

Bibliography 207

Curriculum 243

List of publications 245

List of Figures

1.1 Illustration of principal component analysis 3

1.2 Illustration of blind signal separation 4

1.3 Low-rank matrix factorization 6

1.4 Visualization of a third-order tensor $\mathcal{T}$ 7

1.5 Canonical polyadic decomposition 9

1.6 Multilinear singular value decomposition 9

1.7 Crossing the chasm: tensorization 10

1.8 Reshaping a matrix to a tensor 10

1.9 Overview of the thesis 18

2.1 Organization of the chapter regarding the appearence of tensors. 22

2.2 Tensor data obtained by tensorization as an alternative to natural tensor data or tensor data obtained by experiment design 31

2.3 Evaluation of a trilinear form and bilinear vector function . . . 38

2.4 Natural stacking of polynomial coefficient tensors 42

2.5 Systems of polynomial equations 43

2.6 Examples of solution sets of linear and polynomial equations . 44

2.7 Local and more broader function approximation 45

2.8 Tensorization-based deterministic BSS 55

2.9 Löwner-based tensorization 56

2.10 Comparison between the short-time Fourier transform and continuous wavelet transform 64

2.11 Illustration of the discrete wavelet decomposition 64
2.12 Uniform tree-structured filter bank in a tensor format 66
2.13 Visualization of the stacking of matrices in BSS resulting in a CPD 69
2.14 Illustration of the two procedures yielding the same tensor, based on higher-order derivatives and evaluations 81
2.15 A graph example 85
2.16 Visualization of a neural network with one hidden layer 88

3.1 Decomposition in multilinear rank-$(L_r, L_r, 1)$ terms 97
3.2 Löwner-based tensorization 98
3.3 Löwner-based separation: a toy problem 109
3.4 Influence of noise for Löwner-based separation 110
3.5 Influence of the number of source signals for Löwner-based separation 111
3.6 Löwner-based separation: underdetermined mixture 111
3.7 Influence of the noise and multilinear rank for underdetermined Löwner-based separation 112
3.8 Löwner-based separation: fluorescence experiment 113
3.9 Influence on the number of excitations for Löwner-based fluorescence spectroscopy 114
3.10 Löwner-based FECG separation for a first dataset: observed signals 115
3.11 Löwner-based FECG separation for a first dataset: recovered signals 115
3.12 Löwner-based FECG separation for a second dataset 116

4.1 Segmentation-based tensorization 122
4.2 Segmentation-based function approximation 125
4.3 Segmentation-based separation: a toy problem 134
4.4 Twofold segmentation-based separation: a toy problem 135

4.5 Segmentation-based separation: underdetermined mixture . . . 135
4.6 Influence of noise for segmentation-based separation in the well-conditioned case . . . 136
4.7 Influence of noise for segmentation-based separation in the ill-conditioned case . . . 137
4.8 Influence of the noise for segmentation-based separation in function of the rank-1-ness . . . 138
4.9 Influence of the degree for segmentation-based separation in function of the rank-1-ness . . . 138
4.10 Connection between the number of parameters and relative error for segmentation-based EEG separation . . . 139
4.11 Segmentation-based EEG approximation . . . 140
4.12 Segmentation-based FECG separation . . . 141
4.13 Influence of noise for segmentation-based DOA estimation in the far field case . . . 143
4.14 Influence of noise for segmentation-based and other DOA estimation algorithms in the near field case . . . 143
4.15 Influence of noise for segmentation-based and other DOA estimation algorithms in the small scale case . . . 144

5.1 Constellation diagrams . . . 147
5.2 Comparison of the rank-1 detection algorithm and a basic CPD 165
5.3 Influence of noise for CPSK separation . . . 166
5.4 Influence of the number of samples for CPSK separation . . . 167
5.5 Influence of noise for 16-QAM separation . . . 167
5.6 Illustration of 16-QAM separation . . . 168
5.7 Influence of noise for CPSK deconvolution . . . 168

6.1 Decomposition in multilinear rank-(L_r, M_r, N_r) terms . . . 177
6.2 Structured data fusion . . . 179
6.3 Illustration of variety of factor matrix structures available in SDF 179

6.4 Illustration of contraction in `sdf_nls` 182

7.1 Approximation of two toy functions using nonnegative polynomials 194

7.2 NMF-based hyperspectral imaging: recovered endmember signatures 195

7.3 NMF-based hyperspectral imaging: observed signatures, and recovered aluminum signature from general NMF 195

7.4 NMF-based chemical shift brain imaging 196

7.5 Timing comparison between NP-NMF algorithms and general NMF algorithms 197

List of Tables

2.1 Hankel matrix ranks for different exponential polynomials . . . 47

2.2 Overview of functions that may lead to low-rank matrices/tensors after tensorization . . . 57

4.1 Rank properties of Hankel matrices . . . 125

4.2 Reduction of the number of parameters after segmentation-based compression . . . 128

6.1 Comparison of CCPD and SDF . . . 182

Chapter 1

Introduction

> "**Data is the new oil of the (digital) economy** — Data in the 21st century is like oil in the 18th century: an immensely, untapped valuable asset. Like oil, for those who see data's fundamental value and learn to extract and use it, there will be huge rewards."
>
> — Joris Toonders in Wired [361]

1.1 The new oil

Data is everywhere, and there is quite a lot of it. IDC, a market-research firm, predicts that the amount of data created each year will exceed 180 zettabytes in 2025 [367]. That is 180 followed by 21 zeros. Based on an estimate of 8.5 billion people on Earth in 2025 [369], this amounts to 40 megabytes of data per minute produced per person.

Various types of sources are responsible for this data. Each one of us can be considered to be a data source by just carrying a mobile device, for example. When accessing social media and using search engines, your personal preferences are saved. During commuting, your location is retained. The content of your email box is also stored and analyzed. For many years, we have mainly known the internet as a platform for connecting people. Under the heading of Internet of Things, however, it increasingly interconnects a diversity of physical devices such as mobile and wearable devices, (self-driving) vehicles and various kinds of sensors integrated in health care instruments and home appliances such as thermostats, refrigerators and washers [25]. Besides more diverse types of data, also higher volumes are obtained, yielding the notion of *big data*. While images

taken by the first phone camera in history only consisted of 0.35 megapixels, the resolution of current camera phones can well exceed 40 megapixels [189].

While there might be plenty of oil available underneath the surface, oil only contains value after it is pumped and refined. The same is true for data: it is only valuable if one is able to extract insightful information [353]. The process of *data mining* transforms raw data in valuable information such as patterns or relationships. Techniques from *machine learning* subsequently turn this information into models enabling various predictions, which can be used for economical or technological purposes. Referring to the aforementioned examples, your personal preference and search history are used to send you personalized recommendations and offers, and your location is used to derive a global traffic congestion model. Furthermore, your emails are screened to automatically export meeting arrangements, hotel bookings and flight schedules to your calendar. Image processing tools enable animal detection, allowing self-driving cars to automatically decelerate if needed. Prediction techniques based on mobile healthcare data improve diagnostics and treatments, while sensors in refrigerators or heating elements detect energy inefficiencies and optimize energy use.

The concept of data bearing value has influenced the economy to a large extent. Newly founded companies such as startups are increasingly oriented towards data analytics as opposed to focusing on a physical product. Large firms are frequently seen to pivot to this data-driven world, such as Amazon with its subsidiary Amazon Web Services [226]. Considering social media, Facebook has acquired Instagram in 2012 and WhatsApp in 2014, while LinkedIn is now part of Microsoft. These deals have not occurred because of the recurrent revenue and profit generating capabilities of the smaller firms; the firms are mainly acquired for their data. The recent fine for Facebook imposed by the European Commission for unauthorized use of WhatsApp's data is just one part of the evidence [147].

1.2 One-way and two-way data

Data structures Data representation is central to data mining. Vectors and matrices provide well-known structures to collect, store and represent data. Vectors and matrices consist of one and two modes, respectively, denoting them as one-way and two-way data objects. For example, measuring a physical quantity in function of time yields one-way time series, which can be naturally stored in a vector. The storage of review scores for a movie dataset requires a matrix, containing the score of each user for different movies. A grayscale image

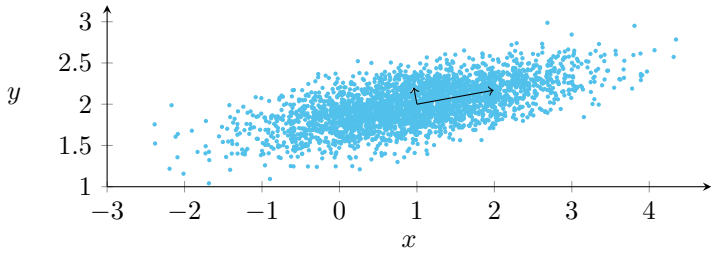


Figure 1.1: Illustration of PCA on a bivariate Gaussian distribution with a clear correlation between the variables x and y. Each vector shown is a normalized principal component scaled by the square root of its corresponding eigenvalue. These principal components indicate the principal modes of variation.

is typically stored as a matrix as well, each entry depicting an image pixel. A graph is a collection of nodes (representing web pages, individual persons or cities, among others) and edges and can also be represented by a matrix. The entry at position (i, j) of the so-called graph adjacency matrix gives information on the connection between nodes i and j.

Data mining tools Many different tools are available to extract valuable information from vector and matrix data. Google's search engine, for example, strongly relies on PageRank which measures the importance of web pages given an adjacency matrix [291]. Movie recommendation systems typically rely on matrix completion tools [38, 66] to predict the unknown entries of a movie rating matrix. Matrix compression techniques are used to reduce the size of image or video data without significantly undermining the quality and interpretation. Principal component analysis (PCA) is another prevalent matrix data mining tool, having significant importance for, e.g., the face recognition algorithm called Eigenfaces [366]. Given a set of observations of possibly linearly correlated variables (the columns of a matrix), PCA derives a specific set of uncorrelated variables called principal components, as illustrated in Figure 1.1 [208].

Before continuing the discussion, let us briefly focus on two other matrix-related techniques, namely blind source separation and clustering.

- Given a set of mixed signals, the goal of blind source separation (BSS) is to recover the original set of source signals without (or as little as possible) prior knowledge of the mixing process or source signals. By stacking the sampled versions of the observed signals in the rows of a matrix $\mathbf{X}$, it can be seen that BSS boils down to the analysis of this matrix. The mixing

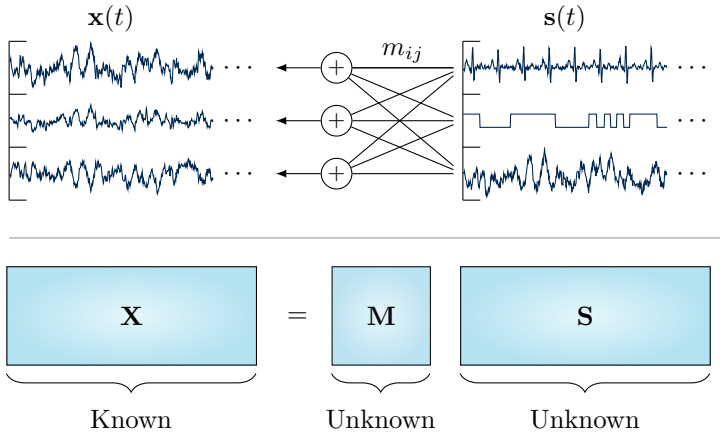


Figure 1.2: Blind signal separation can be seen as a matrix problem. Each observed signal in $\mathbf{x}(t)$ is a linear mixture of the source signals in $\mathbf{s}(t)$. The rows of $\mathbf{X}$ contain sampled versions of the signals in $\mathbf{x}(t)$. Given only $\mathbf{X}$, the goal is to identify both the mixing matrix $\mathbf{M}$ and the source signals in $\mathbf{S}$.

process is often linear such that the model reduces to $\mathbf{X} = \mathbf{MS}$ with mixing matrix $\mathbf{M}$ and set of source signals $\mathbf{S}$, as illustrated in Figure 1.2.

An intuitive example is the cocktail party problem which consists of separating human voices from other speech signals and background music. In a biomedical context, BSS allows separating maternal and fetal electrocardiography (ECG) signals [111] or removing artifacts from electroencephalography (EEG) data [209], among many other applications. BSS is closely related to direction of arrival (DOA) estimation from array processing, while other applications have appeared in telecommunications, chemometrics and in the financial sector [89].

Physical signals such as biomedical or telecommunication signals typically do not propagate instantaneously. Rather than considering an instantaneous mixture as in Figure 1.2, it might be useful to incorporate time delays in the model. One then refers to deconvolution or blind system identification (BSI) [2], dealing with systems with memory such as systems consisting of finite impulse response filters.

- Clustering consists of assigning objects to groups based on similarity. This allows the recognition of patterns in the data or the analysis of network flows, among others. In machine learning, the objects are typically represented by feature vectors. Based on a number of observations, these

vectors can then be stacked in the rows of a matrix $\mathbf{X}$. In the following simple example, two clusters can be easily identified:

$$\begin{matrix} \mathbf{X} \\ \end{matrix} \qquad \begin{matrix} \mathbf{H} \\ \end{matrix} \qquad \begin{matrix} \mathbf{C} \\ \end{matrix}$$

$$\begin{bmatrix} 1.0 & 2.1 & 3.0 \\ 5.9 & 5.1 & 4.1 \\ 5.8 & 5.0 & 3.9 \\ 1.0 & 2.0 & 3.1 \\ 6.1 & 4.8 & 4.0 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \end{bmatrix}$$

The given matrix $\mathbf{X}$ is approximated by $\mathbf{HC}$, in which each row of $\mathbf{H}$ 'selects' a row of $\mathbf{C}$, the latter representing a cluster vector.

Matrix factorization Most of the previously mentioned data mining tools rely on the concept of matrix factorization. The most simple version of matrix factorization is the rank-1 decomposition, writing a given matrix $\mathbf{X}$ as the outer product of two nonzero vectors $\mathbf{a}$ and $\mathbf{b}$, respectively, such that $\mathbf{X} = \mathbf{a}\mathbf{b}^{\mathrm{T}}$. If a matrix $\mathbf{X}$ admits to such a decomposition, it consists of rows (resp., columns) that are scaled versions of each other, such as the following matrix:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \\ -1 & -2 & -3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}.$$

Instead of as a single outer product, the more general rank-R matrix factorization writes $\mathbf{X}$ as a sum of R rank-1 terms:

$$\mathbf{X} = \mathbf{a}_1\mathbf{b}_1^{\mathrm{T}} + \ldots + \mathbf{a}_R\mathbf{b}_R^{\mathrm{T}} = \mathbf{A}\mathbf{B}^{\mathrm{T}},$$

which can be visualized as in Figure 1.3. Given a matrix $\mathbf{X}$ of size $N \times K$, the so-called factor vectors $\mathbf{a}_r$ and $\mathbf{b}_r$ are stacked in the columns of the matrices $\mathbf{A}$ and $\mathbf{B}$ of sizes $N \times R$ and $K \times R$, respectively. If this R is minimal, i.e., if $\mathbf{X}$ cannot be represented by a sum of $R-1$ or less rank-1 terms, this R is defined as the *rank* of the matrix. The theoretical rank of various kinds of matrix data is typically very high, depending on the number of columns and rows. Often, however, a good approximation is obtained if the matrix is approximated by a low number of outer products. The validity of this low-rank matrix factorization follows from the Eckart–Young–Mirsky theorem [142]. It is interesting to see that a total of NK entries in $\mathbf{X}$ are then represented by only $R(N+K)$ entries, e.g., a rank-5 factorization of a matrix of size 100×100 involves only 1000

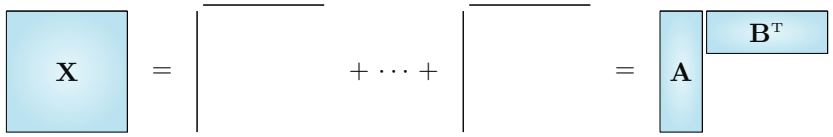


Figure 1.3: A low-rank matrix factorization writes a matrix $\mathbf{X}$ as a sum of outer products of nonzero vectors, stacked in the matrices $\mathbf{A}$ and $\mathbf{B}$.

variables amounting to a reduction of 90%. Such a model complexity reduction provides the foundations of the previously mentioned matrix compression and completion tools. Furthermore, as discussed, the given matrix $\mathbf{X}$ in the BSS and clustering problems is factorized to allow the recovery of the source signals and clustering vectors, respectively. In both cases, each source signal or cluster vector contributes a rank-1 term to $\mathbf{X}$. Finally, dictionary-based learning [264] and factor analysis [182] originating from the domains of machine learning and statistics, respectively, are two topics very related to matrix factorization. Both decompose a given matrix in a set of dictionary vectors or latent factors, respectively.

A fundamental difficulty in the case of low-rank matrix factorization is its non-uniqueness for $R > 1$. Given a matrix $\mathbf{X}$ with factorization $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$, one can always find an invertible matrix $\mathbf{D}$ such that $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}} = \mathbf{A}\mathbf{D}\mathbf{D}^{-1}\mathbf{B}^{\mathrm{T}} = \tilde{\mathbf{A}}\tilde{\mathbf{B}}^{\mathrm{T}}$ with $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{D}$ and $\tilde{\mathbf{B}} = \mathbf{B}\mathbf{D}^{-\mathrm{T}}$. The lack of (essential) uniqueness prohibits a meaningful interpretation of the factor vectors. Additional constraints are typically imposed on $\mathbf{A}$ and/or $\mathbf{B}$ to achieve (essential) uniqueness. Examples of constraints are orthogonality, triangularity and nonnegativity, leading to the singular value decomposition (SVD), LQ or QR decomposition and nonnegative matrix factorization (NMF), respectively. Although useful in various applications, these constraints are not suitable for BSS and clustering. In BSS, independent component analysis (ICA) relies on the assumption of mutually statistically independent source signals in $\mathbf{S}$ [89, 210], while uniform linear sensor arrays in array processing lead to exponential vectors in $\mathbf{M}$ [323]. In clustering, sparse constraints can be applied on $\mathbf{H}$ leading to sparse component analysis (SCA) [60].

1.3 Multiway data

Vectors and matrices consist of one and two mode(s), respectively. One might wonder: why would data science, and nature in general, be limited to two modes? The answer is simple — it is not.

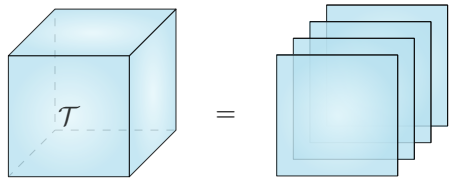


Figure 1.4: Visualization of a third-order tensor $\mathcal{T}$ as a collection of stacked matrices along the third mode. Each matrix can then consist of movie rating scores of users given in a specific year, or represent one of the color coordinates in an RGB-encoded color image.

Data structures Multiway data depend on three or more variables. These data can be naturally stored in multiway arrays consisting of three or more modes, which we define as tensors. Tensors are natural generalizations of vectors and matrices, and the number of modes is defined as its order. Much like a matrix can be seen as a collection of column vectors, a third-order tensor, for example, can be seen as a collection of matrices along its third mode, as illustrated in Figure 1.4. These matrices are defined as the (frontal) slices of the tensor.

Many different data can be understood as multiway data. Measuring physical quantities such as temperature or radiation intensity (think of magnetic resonance imaging (MRI)) along the conventional spatial coordinates readily generates tensor data. Color image data (with the color space coordinates as the third mode) and video data (with the time as third mode) are two types of extensions of gray-scale image data to multiway data. Users typically provide movie ratings at different time instances, yielding the Netflix rating tensor with user, movie and time modes [38]. Furthermore, matrices of the same size depending on a parameter can be stacked into a tensor, e.g., sets of time-dependent graph adjacency matrices [351], mixture-dependent excitation–emission matrices [56] or lag-dependent covariance matrices [33]. One may repeat experiments several times, under varying conditions, with different attributes, etc. Each condition and attribute then refers to a mode.

Data mining tools Tensors representing multiway data have been around since the 19th century. A first step in tensor analysis could be the unfolding of a given tensor to a matrix, after which this matrix is then investigated using matrix tools. Unfolding is defined as the horizontal or vertical stacking of the various slices of a tensor, e.g., a tensor of size $10 \times 10 \times 10$ can be unfolded to a matrix of size 10×100. Eigenfaces, for example, first unfolds a given image to a vector. Hence, given a set of images which can be stacked in a tensor, it

reshapes this tensor first to a matrix, after which PCA is applied [366].

It easily follows that some structure is obfuscated and that in various applications the data multiway characteristics is not entirely exploited. For example, it can be easily seen that video data can be compressed more efficiently by taking into account the temporal dependency as well. Second, a movie recommendation system might attain a higher accuracy when exploiting the additional rating time information. Many of the previously mentioned matrix data mining tools such as PageRank, prediction systems, compression techniques and PCA have therefore been extended to the tensor domain leading to for example multilinear PageRank and multilinear PCA techniques [114, 163].

Tensor decompositions Not surprisingly, many of these extensions are based on tensor generalizations of matrix factorization techniques. Initiated by researchers such as Harshman, Tucker, Kruskal, Carroll and Chang, these tensor decompositions became available from the 70's on, allowing knowledge inference in a compelling way [72, 183, 230]. Multilinear algebra encompasses the theory of tensor tools and tensor decompositions. Rather than just providing extensions of results from linear algebra, multilinear algebra is fundamentally more rich compared to linear algebra. This can be evidenced by the various influential articles and books [84, 177, 223, 229, 233, 324]. Already the concept of rank, which is well defined in the matrix case, deserves a more subtle discussion in the tensor case. For example, the rank of a tensor and the multilinear rank of a tensor are two different concepts, each connected with a different tensor decomposition.

On the one hand, a rank-1 tensor of order N is defined as the outer product of N nonzero vectors. If a tensor can be written as a minimal sum of R such rank-1 terms, its rank is defined as R. Such a decomposition is called the canonical polyadic decomposition (CPD) [72, 183, 230], illustrated in Figure 1.5. Interestingly enough, while we have discussed that low-rank matrix factorization techniques require additional constraints to yield a unique solution, a low-rank CPD is *essentially unique* under mild conditions by itself. Essential uniqueness allows the factor vectors to be determined up to scaling and permutation. It is thus valid to see factor vectors as the *underlying components* of the data. For example, if a tensor consists of stacked excitation–emission matrices of different mixtures, the factor vectors in the first, second and third mode represent the excitation spectrum, emission spectrum and concentrations of the different chemical analytes present in the mixture, respectively [56]. A meaningful solution of the problem at hand is therefore obtained without needing to impose additional constraints. Various results on CPD uniqueness have appeared in the past [134–136, 230, 322].

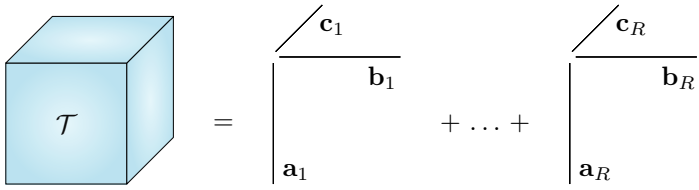


Figure 1.5: Visualization of a CPD of a rank-R third-order tensor. In excitation–emission spectroscopy, each set of vectors $\mathbf{a}_r$, $\mathbf{b}_r$, $\mathbf{c}_r$ can represent the excitation spectrum, emission spectrum and concentrations corresponding to a specific chemical analyte.

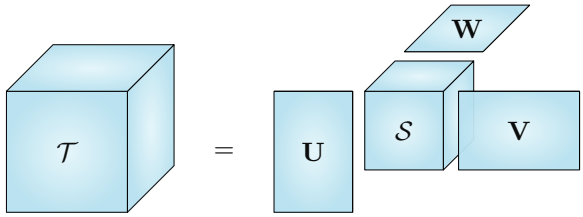


Figure 1.6: Visualization of an MLSVD of a third-order tensor with multilinear rank smaller than its dimensions. The factor matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ represent the mode-n vector spaces while the different interactions between those spaces are contained in the core tensor $\mathcal{S}$.

Denoting the row and column vectors of a matrix or tensor as its mode-1 and mode-2 vectors, a tensor can also be described by the spaces spanned by its mode-n vectors. This yields the multilinear singular value decomposition (MLSVD) [109, 364], as illustrated in Figure 1.6. The tuple consisting of the dimensions of these spaces is defined as the multilinear rank of the tensor. Remarkably, while the dimensions of the row space and column space of a matrix are equal, the mode-n ranks of a tensor can differ. A tensor can typically be well approximated by a truncated MLSVD. The core tensor can then be seen as a compressed version of the original tensor. For fixed dimensions of the core tensor, the truncated MLSVD might not necessarily yield the most optimal approximation. The low multilinear rank approximation (LMLRA) might be better suited [112].

Note that there exist a number of other tensor decompositions as well, such as the decomposition in multilinear rank-$(L_r, L_r, 1)$ terms [101, 345, 346], the block term decomposition [103, 104, 113], the tensor train decomposition [286] and the hierarchical Tucker decomposition [170, 178]. Each of these decompositions might introduce a different definition for the concept of rank.

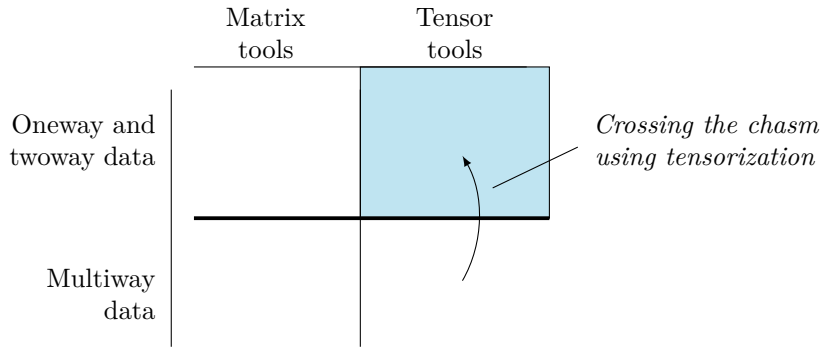


Figure 1.7: Tensorization enables the use of tensor tools on one-way or two-way data.

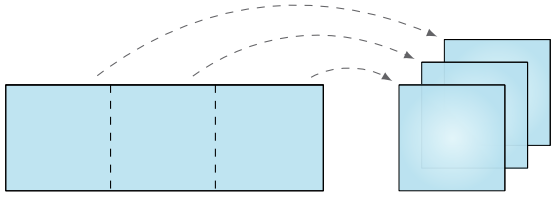


Figure 1.8: An example of tensorization by folding or reshaping a matrix to a tensor, enabling the use of tensor tools on the obtained tensor.

1.4 Crossing the chasm: Tensorization

Let us now recall one of the previous discussions: many matrix tools are available to analyze matrix data and have also been used to cope with multiway data, before tensor tools became available.

Throughout the years, powerful properties and features of tensor mining tools have been revealed. Applying tensor tools obviously requires the availability of a tensor. One might only have a single vector (such as a time series) or matrix (such as an adjacency matrix or grayscale image) available. Nevertheless, a number of methods have appeared in the literature consisting of the application of tensor tools given only a vector or matrix. These methods first map the vector or matrix to a tensor before applying a tensor decomposition or other multiway analysis tool. We refer to this first step as *tensorization*. It allows the use of tensor tools on one-way or two-way data, cf. Figure 1.7. A straightforward example of tensorization is the folding or reshaping of a vector or matrix to a

tensor, as illustrated in Figure 1.8. A matrix of size 3×9, e.g., can then be tensorized to a tensor of size $3 \times 3 \times 3$. Note, however, that there are multiple ways of reshaping a vector or matrix. Given a vector of length 6, for example, two consecutive segments such as $\begin{bmatrix}1 & 2 & 3\end{bmatrix}^{\mathrm{T}}$ and $\begin{bmatrix}4 & 5 & 6\end{bmatrix}^{\mathrm{T}}$ can be stacked in the columns of matrix, or two decimated versions such as $\begin{bmatrix}1 & 3 & 5\end{bmatrix}^{\mathrm{T}}$ and $\begin{bmatrix}2 & 4 & 6\end{bmatrix}^{\mathrm{T}}$ can be stacked:

$$\begin{bmatrix}1 & 2 & 3 & 4 & 5 & 6\end{bmatrix}^{\mathrm{T}} \rightsquigarrow \begin{bmatrix}1 & 4\\ 2 & 5\\ 3 & 6\end{bmatrix} \quad \text{or} \quad \begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix},$$

leading to the techniques of segmentation and decimation, respectively.

Many matrix-related data mining methods such as BSS, BSI and clustering can strongly benefit from tensorization. As discussed, low-rank matrix factorization does not readily enable the recovery of the underlying components, while some tensor decompositions do provide an essentially unique decomposition. By tensorizing the given matrix using a meaningful tensorization technique and subsequently decomposing the obtained tensor using readily available tensor decomposition algorithms, a powerful matrix data mining method can be obtained. Hence, tensorization can be used in various domains such as signal processing, graph analysis, machine learning and chemometrics.

Crossing the chasm A startup is a newly established business with a scalable business model, likely to experience extreme growth[1] soon. Snap(chat), Instagram, Facebook, Apple, Tesla, ... were all startups at one point. These firms started by delivering their products to a number of early adopters who believed in the product. In the startup scene, the expression *crossing the chasm* indicates that a product or service is being bought by a larger market segment, rather than by only a small group of believers [271].

The expression can also be attributed to tensorization. Vector and matrix data are not necessarily more common than tensor data, but still more recognized in scientific and corporate contexts. Rather than only be applied in the 'niche' segment of multiway data, tensorization allows tensor tools to cross the chasm and to be applied on vector or matrix data as well, enabling the latter to benefit from powerful mining tools.

[1] Hence, your favorite bakery or barber is most probably not a startup.

1.5 Research aims

We provide a number of research aims, each clarified subsequently.

> **Aim:** To provide a comprehensive tensorization overview and to develop novel tensorization techniques.

The aforementioned reshaping technique is just one of the many tensorization techniques available in literature. Other techniques use Hankel matrices, are based on higher-order statistics or consider time–frequency or time–scale transforms. Tensorization techniques have appeared rather disparately, and some methods have not even been recognized as tensorization techniques or have not been considered in a tensor setting.

> **Aim:** Given a tensorization technique, to uncover relations between the properties of the original data and the properties of the obtained tensor.

Of course, not every combination of mapping and tensor tool is meaningful. Reshaping a graph adjacency matrix to a tensor and then applying a CPD, for example, might not be worthwhile. Second, while a tensorization can be meaningful for one particular type of vector, it is not necessarily suited for another type of vector. Hence, we have to verify carefully how the tensorization at hand translates the properties of the given vector or matrix to the properties of the obtained tensor. For example, the following two connections have been established earlier:

- An exponential can be mapped to a rank-1 Hankel matrix and, more generally, a rank-1 Hankel tensor [293, 323]. Indeed, let us for example rearrange the vector $\left[1, z, z^2, z^3, z^4\right]^{\mathrm{T}}$ in a Hankel matrix of size 3×3. This matrix can be written as the outer product of two nonzero vectors:

$$\left[1, z, z^2, z^3, z^4\right]^{\mathrm{T}} \rightsquigarrow \begin{bmatrix} 1 & z & z^2 \\ z & z^2 & z^3 \\ z^2 & z^3 & z^4 \end{bmatrix} = \begin{bmatrix} 1 \\ z \\ z^2 \end{bmatrix} \begin{bmatrix} 1 & z & z^2 \end{bmatrix}.$$

- The fourth-order cumulant, which is a fourth-order tensor, of a set of mutually statistically independent signals is approximately diagonal [71].

Besides these low-rank and diagonal properties, respectively, there are a number of other relations available, and many more to be discovered. For example, one

might wonder when the reshaping of a given vector or matrix admits a specific tensor decomposition.

> **Aim:** To provide connections between tensorization, tensor properties and tensor decompositions.

The previously mentioned tensor properties, i.e., low rank and diagonality, allow an obtained tensor from tensorization to be decomposed. However, there are tensorization methods which do not readily yield low-rank tensors or diagonal tensors, but which do enable, after some manipulation, the use of tensor decompositions. One example is the analytical constant modulus algorithm (ACMA) [384].

> **Aim:** To provide proof-of-concepts of various tensorization techniques with a particular focus on BSS.

Tensorization techniques provide little value without a specific objective or application in mind. As discussed, BSS forms an ideal basis for the application of tensorization techniques. It is both challenging and interesting to also investigate underdetermined BSS mixtures. Furthermore, it is worthwhile to consider other applications dealing with the recovery of underlying components, such as excitation–emission spectroscopy or graph clustering.

> **Aim:** To provide uniqueness conditions and pay attention to other working assumptions.

In the case of sampled polynomial source signals in BSS, it can be shown that the mixing matrix and original signals cannot be recovered. However, even in this case, there exist tensorization techniques that map the mixtures of polynomials to tensors with interesting properties. Furthermore, these tensors even admit specific tensor decompositions. One might wonder why the recovery of the original polynomials does not succeed. To solve this apparent paradox, we need to recall that some tensor decompositions, such as a CPD, are essentially unique *under mild conditions*. These conditions do not allow polynomials to be separated.

> **Aim:** To provide ready-to-be-used algorithms suitable for both small-scale and large-scale analysis.

If a matrix is reshaped to a tensor as in Figure 1.8, both the original matrix and

obtained tensor have the same number of entries. However, if a vector of length N is mapped to a tensor of size $N \times N \times N$, the number of entries in the tensor significantly exceeds the original number of data points. For large N, one might not even be able to store this tensor. This is a consequence of the curse of dimensionality, referring to the phenomenon that the number of tensor entries increases exponentially with the number of modes [391]. Hence, algorithms are desired that are able to cope with large (structured) tensors.

On the other hand, the tensorization techniques should still be valuable given only small vectors or matrices, e.g., in BSS with only a limited number of samples available. For example, it is well known that higher-order cumulants suffer from estimation errors [153].

1.6 Outline of the thesis

This brings us to the following chapter-by-chapter outline of the thesis. Chapter 2 can be considered as the glue of the thesis connecting the other chapters, as illustrated in Figure 1.9.

Chapter 2 contains a comprehensive tensorization overview. We review techniques which have appeared in the literature such as Hankelization, higher-order statistics and time–frequency & time–scale techniques [87, 89, 101, 293, 323]. On the other hand, we discuss less-known techniques such as Löwnerization, the stacking of Hessian or Jacobian matrices and the collection of monomial relations [124, 139, 339]. We provide links between different tensorization techniques and connect, for example, time–frequency methods with segmentation, and higher-order statistics with mixed discriminants. This allows the transfer of theoretical results between different techniques; we derive, for example, theoretical low-rank properties for time–frequency methods based on results we obtained in the context of segmentation. Furthermore, it is discussed how tensorization techniques can be applied simultaneously, as has been done for example in CPA–IPA [116].

Each tensorization technique goes hand in hand with one or more applications. We discuss BSS and clustering, while we also focus on parameter estimation and function compression. A large number of scientific domains are brought to the attention such as machine learning, graph analysis and signal processing for telecommunications and biomedical sciences.

Related to tensorization is *tensor recognition*. With tensor recognition, we mean that obtaining a tensor sometimes is equivalent to recognizing the implicit presence of a tensor. A number of fundamental mathematical and engineering concepts can very well be formulated using tensor-based expressions. Multilinear operators such as matrix–matrix multiplication and determinants seem to have clear links with tensors, as well as (multivariate) polynomials. While linearization is related to linear algebra, it follows that function approximations in broader neighborhoods (using for example higher-order Taylor series) can be explained in a tensor context. Other topics are connected to tensors as well, such as global polynomial optimization, multilinear classification and systems of polynomial equations, among others.

Chapter 3 focuses on a tensorization technique called Löwnerization which is based on Löwner matrices. The latter have interesting connections with rational functions. Rational functions can be used to model a variety of smooth curves and signals with both low- and high-varying regions. We discuss low-rank properties and structured decompositions of Löwner matrices.

Furthermore, Löwnerization is applied in the context of BSS. It is discussed that the separation of a number of rational functions after Löwnerization reduces to a decomposition in multilinear rank-$(L_r, L_r, 1)$ terms. We provide uniqueness conditions for this separation, and deliver two different proof-of-concepts. First, Löwnerization is used to separate maternal and fetal ECG signals. Second, it is applied in the context of chemical analyte separation. While classical tensor-based methods require multiple mixtures to separate chemical analytes, we show how a single mixture is sufficient to uncover the underlying components using Löwnerization.

Chapter 4 discusses segmentation in the context of signal compression. In many large-scale applications, signals often admit a compact representation which can be obtained using segmentation and tensor decompositions. We show under which conditions this compactness is available. We provide connections with Hankelization and show that next to the signals that admit to a low-rank tensor decomposition after Hankelization, periodic signals also admit a compact representation, among other types of signals.

Segmentation is also used to separate signals. Besides applying the assumption of compact representation on the source level, it can also be applied on the mixing level, for example in the case of a large number of densely spaced sensors. Higher-order segmentation can provide even a more compact representation which can be seen as a blessing of the curse of dimensionality, and segmentation can be applied on both levels simultaneously. The latter two concepts yield two novel types of tensor

decomposition, namely a (rank-$L_r \otimes$ vector) decomposition and a so-called butterfly decomposition, respectively. The technique is illustrated in two case studies, namely in the context of fetal ECG extraction and in the context of both far- and near-field direction of arrival estimation in array processing.

Chapter 5 provides new algorithms for the separation of multi-modulus signals. The analytical constant modulus algorithm (ACMA) is a well-known algorithm in telecommunications, able to separate constant modulus signals. Each sample s of such a complex-valued signal admits to the expression $|s|^2 = ss^\star = c^2$, with c the constant modulus of the signal. Both 4-QAM and binary phase shift keying (BPSK) signals are constant modulus signals. However, the assumption of constant modulus can be too restrictive for a large set of signals. We propose a variant of the method to enable the separation of multi-modulus signals as well. Multi-modulus signals, such as 16-QAM, can be seen as generalizations of constant modulus signals. They allow the sample moduli to be equal to two or more constant moduli rather than only a single constant modulus.

The proposed method transforms the problem of multi-modulus separation algebraically into a set of coupled tensor decompositions. An exact solution is guaranteed by a matrix eigenvalue decomposition in the noiseless case. The standard method requires a minimum number of available samples. A reduction of this bound is obtained by generalizing a previously developed rank-1 detection procedure [70, 99]. This generalization allows the recovery of Kronecker-structured vectors from a given space spanned by both Kronecker-structured vectors and arbitrary vectors.

We extensively compare our method against other methods such as ICA, ACMA and other multi-modulus algorithms [316, 317]. Besides in an instantaneous BSS context, we also showcase our method in a convolutive BSI context.

Chapter 6 groups recent developments in numerical optimization-based computations of tensor decompositions that have formed the basis of the third major release of Tensorlab, of which the candidate was one of the developers. In Tensorlab 3.0, a framework of tensorization techniques is introduced. Furthermore, it is explained how the structure of tensor decompositions can be exploited to obtain both fast and well-converging algorithms, and how coupled factorizations and structured factors can be dealt with. The latter two concepts form the basis of structured data fusion (SDF), which is a framework within Tensorlab allowing the rapid prototyping of analysis and knowledge discovery in one or more tensors.

A tough challenge is how to analyze large-scale datasets, corresponding to tensors with large dimensions or with a large number of modes. The use

of incomplete tensors and randomized block sampling are two possible approaches [388, 391, 393], the latter deliberately dealing with only a small random set of tensor entries. On the other hand, tensorization of a vector matrix can introduce redundancy into the tensorized form. It can be worthwhile to exploit this structure and to only deal with efficient representations of the tensor. With the presented algorithms, we arrive at *implicit tensorization*. The latter describes the process of tensorization combined with tensor decompositions without the explicit construction of a tensor. The benefits of multilinear algebra are retained while some disadvantages, such as the curse of dimensionality, are avoided.

Chapter 7 proposes a novel method for nonnegative matrix factorization (NMF) based on nonnegative polynomials. We will show how a nonnegative polynomial in a finite interval can be formulated using a closed-form expression based on a sum of squares [237]. As polynomials can approximate a wide range of shapes, they can be well suited to model the factor vectors in NMF. This results in an optimization problem without external nonnegativity constraints (which can be solved using conventional optimization techniques), as well as in a significant reduction of the number of variables. Furthermore, the polynomial-based model may realize an intrinsic noise reduction and typically yields smooth results without the need of external smoothing constraints.

In large-scale tensor decompositions, a widely used approach is to first apply a compression using for example an LMLRA. In the matrix case, a compression can be obtained with an SVD. However, in an NMF context, an SVD destroys the nonnegativity in both the data and the model. We show how polynomial-based NMF does allow an orthogonal compression without sacrificing accuracy. This enables the technique to scale well to large matrices. The polynomial-based NMF technique is illustrated with applications in hyperspectral imaging and chemical shift brain imaging.

Chapter 8 finally summarizes the findings of the thesis and suggests directions for prospective work.

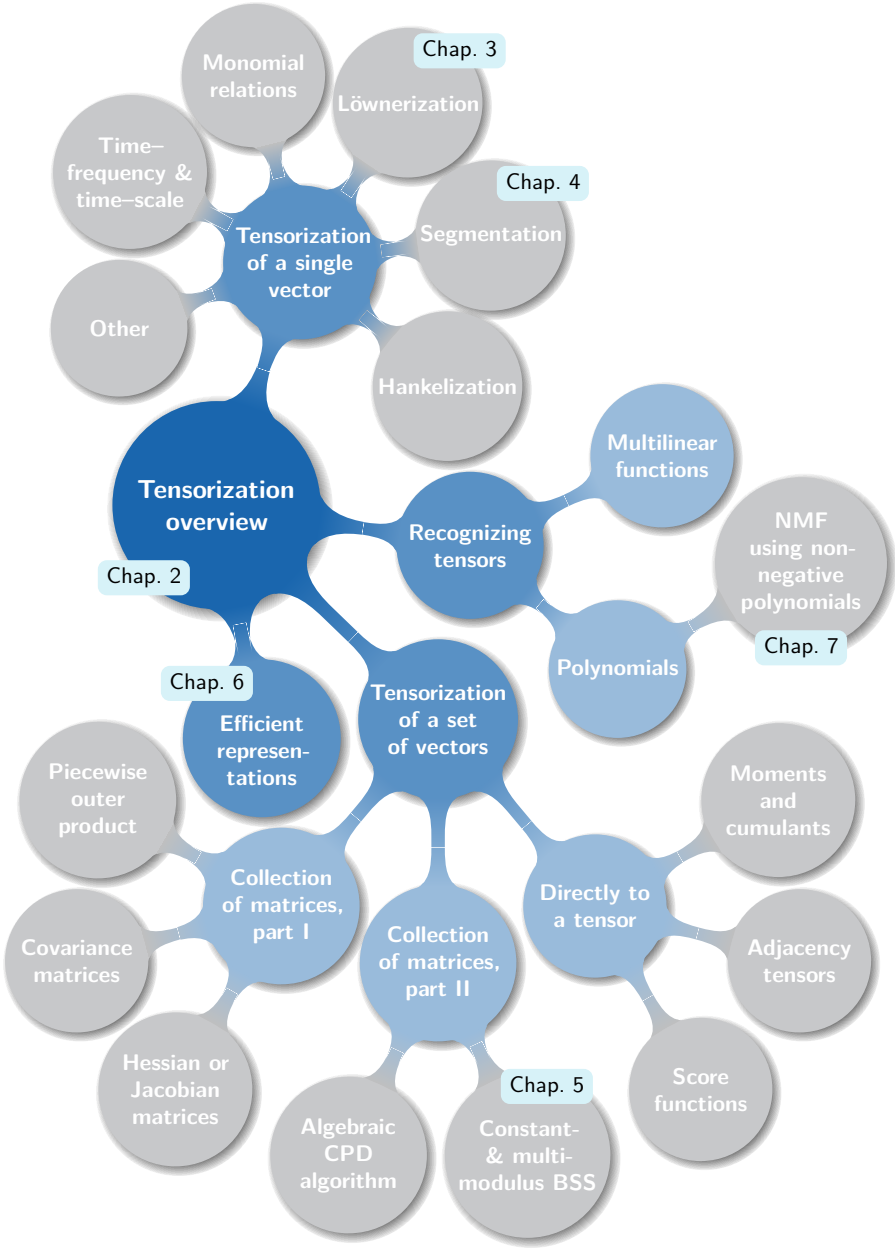


Figure 1.9: Overview of the thesis

Chapter 2

The concept of tensorization

Abstract Multiway arrays, also known as tensors, are ubiquitous in domains such as data mining and signal processing. The powerful dimension reduction properties and uniqueness properties of tensor tools, to name a few, have seen a wide applicability with respect to, e.g., component retrieval and compression. Still, a tensor is required to enable the use of tensor tools. While often only vector or matrix data is available, one can first map the vector or matrix data to a tensor. Various techniques have made use of such a tensorization step, whether or not knowingly. Rather than focusing on tensor tools, we will zoom in on this concept of tensorization. We present an overview of various tensorization techniques and show how tensorization is fundamental in applications such as independent component analysis, neural network training, time–frequency analysis and graph clustering. Furthermore, we illustrate how conceptual problems from engineering and mathematics can be well expressed in a tensor-based manner, such as the approximation of non-linear functions with polynomials, matrix–matrix multiplication and systems of polynomial equations.

Reference This chapter is a slightly adapted version of the article [120]. The candidate performed the research and wrote the article under the guidance of the coauthor.

2.1 Introduction

Tensor tools have been applied widely across various domains such as scientific computing, signal and image processing, machine learning and statistics. Tensors are generalizations of vectors and matrices and provide natural structures to store and represent multiway data. Multilinear algebra forms the mathematical backbone of tensors and tensor tools, encompassing far more rich concepts compared to its linear counterpart. This has allowed multilinear tensor models to surpass the flat-view bilinear matrix models and has enabled novel applications dealing with processes such as component retrieval, compression, classification and prediction. While researchers such as Hitchcock [190, 191] and Cattell [73, 74] have provided the foundations of multilinear algebra in the early 20th century, the spark in interest only came from the 60's on thanks to results attributed to Tucker [364], Harshman [183, 184], Carroll and Chang [72] and Kruskal [230] which mainly originated from the domains of psychometrics and chemometrics.

Various influential survey papers and books have appeared discussing models, algorithms and applications of tensor tools, e.g., [84, 223, 229, 324, 328]. A large class of tensor tools consists of tensor decompositions such as the canonical polyadic decomposition (CPD), the tensor train decomposition (TT) and the multilinear singular value decomposition (MLSVD) or Tucker decomposition. These tensor tools obviously require a tensor. Often, a tensor is readily available, whether in a natural way (e.g., a physical quantity measured along the three common spatial coordinates) or by experiment design (e.g., by repeating an experiment several times and by stacking the obtained data matrices). In some cases, however, only a single vector or matrix is available. Still, an increasing number of techniques from various scientific domains are appearing in which tensor tools are applied given only vector or matrix data. These techniques have been used to separate independent signals [33, 71] or constant modulus signals [384] in signal processing, to identify graph clusters in graph analysis [40], to assign topics to documents in topic modeling [18, 193] and to train neural networks in machine learning [88, 204], to name only a few examples. In each of these applications, only a single vector or matrix is available. Instead of solely being adopted by multiway data, the power and flexibility of tensor tools have seen their way to one-way and two-way data as well.

One might wonder how this apparent paradox can be explained and how tensor tools can be applied given only vector or matrix data. Fundamental in these techniques is the paradigm of mapping the given vector or matrix data to a tensor first, yielding a *tensorization* step in which the lower-order data is tensorized to a tensor. This tensorization procedure is often not well described

and sometimes not even recognized. Tensorization effectively translates the assumed vector or matrix model to a multiway model. The obtained tensor typically admits properties such as low-rankness or diagonality for specific sets of vectors or matrices. Various tensor tools build on such tensor properties. For example, given a set of mutually statistically independent signals, the corresponding fourth-order cumulant is diagonal which allows the use of the CPD in independent component analysis (ICA) [71]. Hence, an investigation of the applied tensorization technique is crucial to be able to validate the use of tensor tools in the subsequent step.

In this overview chapter, we provide a tensorization framework and we survey different tensorization techniques. The goal consists of presenting an insightful view on the concept of tensorization, illustrating the power of tensor tools given only vector or matrix data. Results regarding tensorization have appeared rather disparately in the literature. We will connect different tensorization techniques and show that theoretical results and algorithms can be interchanged between these techniques and, thereby, between scientific domains. For example, as we will see, time–frequency and time–scale representations [87, 97, 265] from signal analysis on the one hand, and segmentation from array processing and function approximation [51, 214] on the other hand, have common grounds such that results from the latter can be passed to the former. Throughout the chapter, we will show how the different techniques can be used in various applications such as deterministic blind signal separation, topic modeling, graph clustering, independent component analysis and neural network training.

Together with tensor construction often comes the recognition of a tensor. With tensor recognition, we mean that obtaining a tensor sometimes boils down to recognizing that a problem already involves a tensor. The tensor may be there only implicitly, and seeing it is not always obvious. We will see that multilinear functions and polynomials can be uniquely represented by a tensor. It follows that a number of fundamental mathematical and engineering concepts can very well be formulated using tensor-based expressions. For example, while the concept of linearization is related to linear algebra, the polynomial-based approximation of a non-linear function corresponds to multilinear algebra. Furthermore, topics such as matrix–matrix multiplication, global polynomial optimization and systems of polynomial equations all have clear links with tensors, as will be discussed. The link between multilinear functions and tensors even allows us to develop a framework based on a process called polarization from which new tensorization techniques can originate.

The chapter is organized as follows. In Section 2.2, a number of preliminaries are provided. Section 2.3 then discusses two common ways how measurements or evaluations readily yield multiway data which can be represented by a tensor. It is clear that multiway data can often be seen as evaluations of multivariate

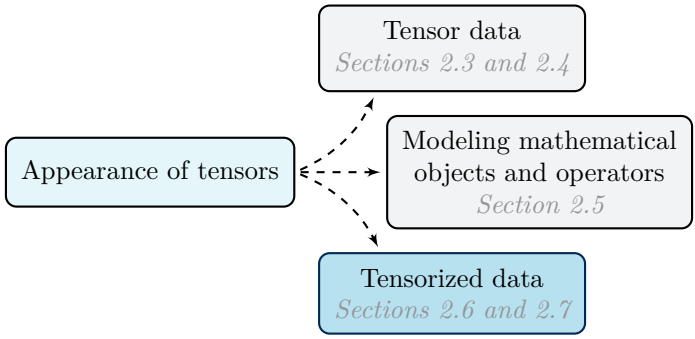


Figure 2.1: Organization of the chapter regarding the appearance of tensors.

functions; the latter are discussed in Section 2.4. Together with function evaluation, also function differentiation is covered. Instead of considering tensors modeling multiway data, Section 2.5 considers tensors representing objects or operators such as multilinear functions and polynomials. The two latter sections form the theoretical spine of tensorization techniques, which are reviewed in Sections 2.6 and 2.7. A distinction is made between those tensorization techniques that are able to map a single vector to a tensor (Section 2.6), and those techniques that map a given matrix to a tensor (Section 2.7). Section 2.8 then provides further discussions and pointers to future work. Figure 2.1 illustrates the organization of the chapter.

2.2 Preliminaries on (multi)linear algebra and multiway analysis

Basic notations are given in Section 2.2.1 while preliminaries on functions are discussed in Section 2.2.2. Matrix factorizations and tensor decompositions are then covered in Sections 2.2.3 and 2.2.4, respectively.

2.2.1 Basic notation and matrix/tensor operations

We will denote scalars, vectors, matrices and tensors by lower case (e.g., a), bold lower case (e.g., $\mathbf{a}$), bold upper case (e.g., $\mathbf{A}$) and calligraphic (e.g., $\mathcal{A}$) letters, respectively. Functions are denoted by underlined letters, e.g., a scalar function $\underline{f}$ and vector function $\underline{\mathbf{f}}$. An Nth-order tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ is a multiway array with numerical values $a_{i_1 i_2 \cdots i_N} = \mathcal{A}(i_1, i_2, \ldots, i_N)$, with $\mathbb{K}$

denoting either $\mathbb{R}$ or $\mathbb{C}$. The mode-n vectors of $\mathcal{A}$ are constructed by fixing all but one index. The ith column of a matrix $\mathbf{A}$ is denoted with $\mathbf{a}_i$. Sets are indexed by superscripts[1] within parentheses, e.g., $\left\{\mathbf{A}^{(n)}\right\}_{n=1}^{N}$. The mode-$n$ unfolding of $\mathcal{A}$ is a matrix $\mathbf{A}_{(n)}$ with the mode-n vectors as its columns. The vectorization operator $\operatorname{vec}(\mathcal{A})$ stacks all mode-1 vectors into a column vector. On the other hand, a vector $\mathbf{a} \in \mathbb{K}^N$ or matrix $\mathbf{A} \in \mathbb{K}^{M \times N}$ can also be folded to a tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times \ldots I_D}$ with $N = \prod_{d=1}^{D} I_d$ and $MN = \prod_{d=1}^{D} I_d$, respectively.

The complex conjugate, transpose, conjugated transpose, inverse, transposed inverse and pseudo-inverse are denoted by $\cdot^\star$, $\cdot^{\mathrm{T}}$, $\cdot^{\mathrm{H}}$, $\cdot^{-1}$, $\cdot^{-\mathrm{T}}$ and $\cdot^\dagger$, respectively. $r(\mathbf{A})$ returns the rank of $\mathbf{A}$ while $\det(\mathbf{A})$ returns the determinant of a square matrix $\mathbf{A}$. The column-wise and row-wise concatenation of vectors and/or matrices $\mathbf{A}$ and $\mathbf{B}$ is defined by $\mathbf{X} = \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}$ and $\mathbf{X} = \begin{bmatrix}\mathbf{A}; & \mathbf{B}\end{bmatrix}$, respectively, with the latter being equal to $\mathbf{X} = \begin{bmatrix}\mathbf{A}^{\mathrm{T}} & \mathbf{B}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$. The vectors $\mathbf{1}_N$ and $\mathbf{0}_N$ denote the $N \times 1$ vectors of all ones and zeros, respectively. A similar definition holds for the matrices $\mathbf{1}_{M \times N}$ and $\mathbf{0}_{M \times N}$ of size $M \times N$. $\mathbf{I}_N$ is defined as the identity matrix of size $N \times N$. The canonical basis vector $\mathbf{e}_i$ contains a one in the ith entry and zero elsewhere.

The inner product between two tensors $\mathcal{A}$ and $\mathcal{B}$ is denoted by $\langle \mathcal{A}, \mathcal{B} \rangle = \operatorname{vec}(\mathcal{B})^{\mathrm{H}} \operatorname{vec}(\mathcal{A})$. The tensor product or outer product of two tensors $\mathcal{A} \in \mathbb{K}^{I_1 \times \cdots \times I_M}$ and $\mathcal{B} \in \mathbb{K}^{J_1 \times \cdots \times J_N}$ is the tensor $\mathcal{C} = \mathcal{A} \otimes \mathcal{B} \in \mathbb{K}^{I_1 \times \cdots \times I_M \times J_1 \times \cdots \times J_N}$ with entries $c_{i_1 \ldots i_M j_1 \ldots j_N} = a_{i_1 \ldots i_M} b_{j_1 \ldots j_N}$ for all indices. The Hadamard or element-wise product and the Kronecker, column-wise Khatri–Rao and row-wise Khatri–Rao products are denoted with $*$, $\otimes$, $\odot$ and $\odot^{\mathrm{T}}$, respectively. The columns of $\mathbf{A} \odot \mathbf{B}$ (resp., rows of $\mathbf{A} \odot^{\mathrm{T}} \mathbf{B}$) are the pairwise Kronecker products of the columns (resp., rows) of $\mathbf{A}$ and $\mathbf{B}$. Note that the outer and Kronecker product are related through a vectorization: $\operatorname{vec}(\mathbf{a} \otimes \mathbf{b}) = \mathbf{b} \otimes \mathbf{a}$. The mode-$n$ product of a tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times \cdots I_N}$ and a matrix $\mathbf{B} \in \mathbb{K}^{J_n \times I_n}$ results in the tensor $\mathcal{C} = \mathcal{A} \cdot_n \mathbf{B} \in \mathbb{K}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$ with entries $c_{i_1 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 \cdots i_N} b_{j_n i_n}$. It can be seen that $\mathcal{C}$ is equal to a folded version of $\mathbf{B}\mathbf{A}_{(n)}$. A mode-$n$ contraction or tensor–vector product of a tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times \cdots \times I_N}$ and a vector $\mathbf{b} \in \mathbb{K}^{I_n}$ is defined as $\mathcal{A} \cdot_n \mathbf{b}^{\mathrm{T}} \in \mathbb{K}^{I_1 \times \cdots I_{n-1} \times I_{n+1} \times \cdots \times I_N}$.

2.2.2 Preliminaries on functions

A scalar(-valued) univariate function, e.g., $\underline{f}(u) = 2u$, maps a single scalar variable to another scalar. Multivariate functions depend on several variables which can be scalars, vectors or tensors such as $\underline{f}(u, v) = u + v$ or $\underline{f}(\mathbf{u}, \mathbf{v}) =$

[1] Superscripts typically indicate a set of objects of different types (such as factor matrices with different dimensions), while subscripts typically indicate a set of objects of similar types which can be naturally stacked (such as vectors of equal dimensions).

$\langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{K}$. On the other hand, functions can also have vectors in their range rather than scalars, such as $\underline{\mathbf{f}}(u) = \left[1, u, \ldots, u^I\right]^{\mathrm{T}} \in \mathbb{K}^I$. Further generalizations consist of matrix(-valued) and tensor(-valued) functions[2]. Vector, matrix and tensor functions can be seen as the natural (higher-order) stacking of scalar functions. Tensor-valued multivariate functions then merge both extensions, such as the outer product function $\underline{\mathcal{F}}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$.

2.2.3 Matrix factorization

Let us consider the following bilinear matrix factorization writing $\mathbf{X} \in \mathbb{K}^{N \times K}$ as a sum of R outer products of nonzero factor vectors:

$$\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}} = \sum\nolimits_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r, \tag{2.1}$$

with $\mathbf{A} \in \mathbb{K}^{N \times R}$ and $\mathbf{B} \in \mathbb{K}^{K \times R}$. The minimal number of rank-1 terms needed to compose $\mathbf{X}$ is called the rank of $\mathbf{X}$.

The decomposition in (2.1) is not unique as one can find a nonsingular matrix $\mathbf{Z} \in \mathbb{K}^{R \times R}$ such that $\mathbf{A}\mathbf{B}^{\mathrm{T}} = \mathbf{A}\mathbf{Z}\mathbf{Z}^{-1}\mathbf{B}^{\mathrm{T}} = \tilde{\mathbf{A}}\tilde{\mathbf{B}}^{\mathrm{T}}$ with $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{Z}$ and $\tilde{\mathbf{B}} = \mathbf{B}\mathbf{Z}^{-\mathrm{T}}$. *Essential uniqueness* implies that the factor vectors can be uniquely determined up to scaling and permutation. Unless $R = 1$, the decomposition in (2.1) is only essentially unique if, additionally, constraints on $\mathbf{A}$ and/or $\mathbf{B}$ are imposed. Well-known constraints include orthogonality, triangularity, Vandermonde and nonnegativity, which — whether individually or combined — form the basis of many fundamental factorizations such as the singular value decomposition (SVD), QR decomposition, Vandermonde decomposition and non-negative matrix factorization (NMF).

Insert 1 — Component recovery in applications: The bilinear model in (2.1) forms the basis of various (generic) applications. In some, such as in matrix completion or compression, the unique recovery of the factor vectors is of subordinate importance to the approximation accuracy [66]. In others, this recovery is critical:

- In blind signal separation, for example, the goal is to recover the source signals in $\mathbf{A}$ and/or the mixing matrix $\mathbf{B}$ given only the observed signals in $\mathbf{X}$. Additional assumptions include mutual independence of the source signals, or Vandermonde structure of the mixing matrix when dealing with uniform linear arrays.

[2]The terminology should not be confused with the popular class of matrix functions such as the matrix exponential [165]. According to the terminology in this chapter, this would be called a matrix-valued function with a matrix-valued variable.

- The goal of (hard) clustering consists of the allocation of objects into different groups or clusters [201]. In machine learning, these objects and clusters are typically represented by feature vectors and cluster vectors, respectively. By stacking the feature vectors in the rows of the matrix $\mathbf{X}$ and assuming zero-variance clusters, one can write $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$. The columns of $\mathbf{B}$ consist of the cluster vectors while each row of $\mathbf{A}$ is a canonical basis vector.
- In spectroscopy, $\mathbf{X}$ is an excitation–emission matrix of a chemical mixture. Each factor vector indicates the (scaled) excitation or emission spectrum of a single chemical component.

Each source signal, cluster or chemical component contributes a rank-1 term to the matrix $\mathbf{X}$. These applications, among others, are covered in the remainder of this chapter under the generic notion of *unique matrix factorization*. Tensor-based methods seem well suited for this problem due to their interesting properties, as will be discussed.

2.2.4 (Coupled) Tensor decompositions

We briefly recall a number of basic concepts from multilinear algebra such as (multilinear) rank and tensor decompositions. For more details, we refer the reader to the provided pointers or to accessible survey papers such as [84, 223, 324].

Rank and decomposition in rank-1 terms

An Nth-order tensor $\mathcal{T}$ has rank one if it can be written as the outer product of N nonzero vectors. The (canonical) polyadic decomposition (CPD) writes $\mathcal{T} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ as a (minimal) sum of R rank-1 tensors:

$$\begin{aligned} \mathcal{T} &= \sum\nolimits_{r=1}^{R} \mathbf{u}_r^{(1)} \otimes \mathbf{u}_r^{(2)} \otimes \cdots \otimes \mathbf{u}_r^{(N)} \\ &\triangleq \left[\!\!\left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)} \right]\!\!\right]. \end{aligned} \tag{2.2}$$

If each rank-1 term is weighted with a factor c_r, the notation $\left[\!\!\left[\mathbf{c}; \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)} \right]\!\!\right]$ is used.

Generalizing the definition of matrix rank, the rank of $\mathcal{T}$ is defined as the minimal number of rank-1 terms needed to compose $\mathcal{T}$. Unlike in the matrix case, the rank of a tensor can exceed its dimensions. A tensor of size $3 \times 3 \times 3$, for example, can have rank 4.

The CPD in (2.2) is essentially unique if the factor vectors can be recovered up to permutation of the rank-1 terms, and up to scaling and counterscaling of the factors in the same rank-1 term. In the tensor case, essential uniqueness is expected under only mild conditions without needing to impose constraints on the factor matrices. These conditions can be deterministic (given a particular set of factor matrices) or generic (assuming factor entries drawn from a continuous probability density function); a concise tutorial is given in [324, Section IV]. Kruskal's condition is a well-known deterministic uniqueness condition and states that the rank-R CPD $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \in \mathbb{K}^{I \times J \times K}$ with $\mathbf{A} \in \mathbb{K}^{I \times R}$, $\mathbf{B} \in \mathbb{K}^{J \times R}$ and $\mathbf{C} \in \mathbb{K}^{K \times R}$ is essentially unique if $k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2R + 2$ [230], with the Kruskal rank $k_{\mathbf{M}}$ of a matrix $\mathbf{M}$ defined as the largest integer k such that any k columns of $\mathbf{M}$ are linearly independent. On the other hand, the following is an example of a generic uniqueness condition of a rank-R third-order tensor $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ [79]:[3]

$$R \leq \left\lceil \frac{IJK}{I + J + K - 2} \right\rceil - 1 \text{ and } IJK \leq 15000,$$

with $\lceil x \rceil$ denoting the smallest integer not less than x. In the case of $I = J = K = 10$, the rank R can be up to 35 for generic uniqueness. Other deterministic and generic conditions (and order-N generalizations) have been developed in [134–136, 322, 345] and [78, 131, 348], respectively.

This essential uniqueness is only one of the foundations of the success of tensor-based methods and is fundamentally different from essential uniqueness in the matrix case. It allows unique signal separation and a meaningful extraction of features or components, among others. Furthermore, unlike in the matrix case, we have seen that the rank of a tensor can exceed its dimensions. This allows for example the recovery of source signals in an underdetermined setting, i.e., when there are fewer observed signals than source signals. Both aforementioned examples of uniqueness conditions are rather mild, meaning that they are realistic working assumptions in applications.

Let us also briefly consider simultaneous matrix diagonalization. Given a set of K matrices $\mathbf{X}_k \in \mathbb{K}^{I \times J}$, the matrices $\mathbf{A} \in \mathbb{K}^{I \times R}, \mathbf{B} \in \mathbb{K}^{J \times R}$ are then found

[3] While expected to hold for larger number of entries IJK as well, the condition has only been verified numerically up to 15000 entries.

such that

$$\mathbf{X}_1 = \mathbf{A}\mathbf{\Lambda}_1\mathbf{B}^{\mathrm{T}},$$

$$\vdots$$

$$\mathbf{X}_K = \mathbf{A}\mathbf{\Lambda}_K\mathbf{B}^{\mathrm{T}},$$

with diagonal matrices $\mathbf{\Lambda}_k \in \mathbb{K}^{R\times R}$. Let $\mathbf{X}_k$ be the kth slice of a tensor $\mathcal{X} \in \mathbb{K}^{I\times J\times K}$. It can be seen that the simultaneous matrix diagonalization problem is equivalent with finding the CPD $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ with the diagonals of the matrices $\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_K$ stacked in the rows of $\mathbf{C} \in \mathbb{K}^{K\times R}$. Hence, it is worthwhile to interpret simultaneous matrix diagonalization problems and techniques (such as [33, 71, 175, 384]) in a tensor-based setting, from which algorithms and uniqueness/identifiability conditions are readily available.

Multilinear rank, low multilinear rank decomposition and block term decomposition

The mode-n rank of a tensor $\mathcal{T} \in \mathbb{K}^{I_1\times I_2\times\cdots\times I_N}$ is defined as the rank of the mode-n unfolding $\mathbf{T}_{(n)}$. The multilinear rank of $\mathcal{T}$ is then defined as the tuple of mode-n rank values. Unlike in the matrix case, the mode-n rank values can differ between modes. A multilinear singular value decomposition (MLSVD) writes a multilinear rank-$(R_1, R_2, \ldots, R_N)$ tensor $\mathcal{T}$ as the tensor-matrix product of an ordered all-orthogonal core tensor $\mathcal{S} \in \mathbb{C}^{R_1\times R_2\times\cdots\times R_N}$ with orthogonal factor matrices $\mathbf{U}^{(n)} \in \mathbb{K}^{I_n\times R_n}$ for $1 \leq n \leq N$. In practice, a tensor can often be well approximated by a truncated MLSVD with small core dimensions R_n. However, for fixed R_n, the low multilinear rank approximation (LMLRA) might yield a better approximation [112]. Because of these properties, the LMLRA and MLSVD are popular for compression and denoising purposes [114, 326].

Block term decomposition, decomposition in multilinear rank-$(L_r, L_r, 1)$ terms and other tools

A block term decomposition (BTD) writes a tensor as a sum of R low multilinear rank terms [103, 104, 113]. For a third-order tensor $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$, we have

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{G}^{(r)} \bullet_1 \mathbf{A}^{(r)} \bullet_2 \mathbf{B}^{(r)} \bullet_3 \mathbf{C}^{(r)},$$

with $\mathcal{G}^{(r)} \in \mathbb{K}^{L_r\times M_r\times N_R}$, $\mathbf{A}^{(r)} \in \mathbb{K}^{I\times L_r}$, $\mathbf{B}^{(r)} \in \mathbb{K}^{J\times M_r}$ and $\mathbf{C}^{(r)} \in \mathbb{K}^{K\times N_r}$. Each term has multilinear rank-(L_r, M_r, N_r). A special case is the BTD in

multilinear rank-$(L_r, L_r, 1)$ terms [101], writing $\mathcal{T}$ as:

$$\mathcal{T} = \sum_{r=1}^{R} \left(\mathbf{A}^{(r)} \mathbf{B}^{(r)^{\mathrm{T}}} \right) \otimes \mathbf{c}_r.$$

These terms are more general than rank-1 terms in a CPD. Nevertheless, it has been shown that the BTD in multilinear rank-$(L_r, L_r, 1)$ terms is still unique under mild conditions [101, 345].

A decomposition in (rank-L_r $\otimes$ rank-M_r) terms writes a tensor $\mathcal{T}$ as a sum of R terms, each consisting of the outer product of a rank-L_r tensor and a rank-M_r tensor [51].

The tensor train (TT) decomposition represents a higher-order tensor as a set of matrices and third-order tensors [286]. It is a particular type of tensor network [283], as well as the hierarchical Tucker (hT) decomposition [170, 177].

Structured and coupled tensor decompositions

Structured tensor decompositions impose additional constraints or structure on the factor matrices, such as orthogonality, nonnegativity or Hankel structure [333]. These constraints can implement application-dependent prior knowledge. Although typically not necessary for essential uniqueness for sufficiently low rank, they can enforce uniqueness even if the rank significantly exceeds the tensor dimensions [337]. Furthermore, exploiting structure can be beneficial for interpretability, well-posedness and/or accuracy.

Coupled tensor decompositions consider the decomposition of a set of tensors with coupled/shared factor matrices. They provide necessary tools within the concept of data fusion [6, 232, 333]. For example, given two datasets in the form of third-order tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$, the goal could be to find matrices $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(5)}$ such that $\mathcal{T}^{(1)} = \left[\!\left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(5)} \right]\!\right]$ and $\mathcal{T}^{(2)} = \left[\!\left[\mathbf{U}^{(3)}, \mathbf{U}^{(4)}, \mathbf{U}^{(5)} \right]\!\right]$ with a coupled factor matrix $\mathbf{U}^{(5)}$ in the third mode. Relaxed uniqueness conditions can be one of the motivations for the use of coupled decompositions. Algorithms and uniqueness results can be found in [6, 340, 407] for coupled CPDs (cCPDs) and in [345, 346] for the coupled decomposition in multilinear rank-$(L_r, L_r, 1)$ terms, among others. A general coupling model considering different types of tensor decompositions can be found in [333].

Tensor decomposition algorithms

The typical algorithms are either algebraic or optimization-based. Examples of the former class build on (generalized) matrix eigenvalue decompositions (GEVD) and can be computed with a polynomial time complexity, e.g., see [99, 130, 136, 242, 312] for the CPD case. If their working conditions for exact decompositions are satisfied, the algebraic algorithms are guaranteed to return exact factors. In the inexact case, e.g., in the presence of noise, they are suboptimal but they can provide good starting values for optimization-based algorithms. Algorithms of the latter class are based on, e.g., alternating optimization techniques, all-at-once methods or stochastic gradient approaches. These algorithms try to minimize the differences between the given tensor and the purported model. Examples for the CPD case are [72, 183, 297] and [8, 332] and [393], respectively. Various toolboxes are available such as Tensorlab [392], the N-way toolbox [20] and the Tensor Toolbox [28], among others.

2.3 Multiway experiment design

Throughout the years, many powerful tensor tools have been developed to cope with the inherent multiway character of datasets. The acquisition of multiway datasets is an important first step before tensor tools can be applied. Frequently, as will be discussed in this section, experiments readily generate tensor data. In other cases, experiments only yield a single data vector or data matrix. Using tensorization, which will be covered in the remainder of this chapter, this vector or matrix data are then transformed to tensors outside of the data creation process. This is illustrated in Figure 2.2.

Some data obtained from experiments have a natural multiway character. For example, measuring physical quantities such as temperature, volumetric mass density or radiation intensity (think of magnetic resonance imaging (MRI)) along the conventional spatial coordinates readily generates tensor data [410]. A physical quantity of a (chemical) mixture can also be expressed in a tensor of which the modes correspond to the concentrations of the individual components [391]. As will be discussed in Section 2.4, such datasets can be seen as evaluated scalar multivariate functions [171, 177, 214], e.g., $\underline{f}(x, y, z)$ for tensors with three spatial modes.

Likewise, the entry at position (i, j, k) of a tensor can indicate, for example, whether subjects i, j and k participated in the same social network discussion [7], giving rise to multiway relational data. Given N objects, each entry of a

third-order similarity tensor of dimensions $N \times N \times N$ represents the similarity between three of those objects.

In the previous examples, the modes share a similar meaning. However, this is not necessary. While a grayscale image can be associated with a matrix, a color image can be perceived as tensor data when stacking the coordinates in the color coding space (such as RGB or CMYK) along a third mode [255, 409]. The first two modes then correspond to the spatial coordinates of the image. Hyperspectral imaging extends color imaging by collecting many more measurements along the third spectral mode [76, 262]. Video data is a natural extension of image data, including a time mode in addition to the pixel and spectral modes [219, 261].

Sometimes one deliberately includes an additional variable in the data creation, allowing for example the creation of tensor data when otherwise only matrix data would be available. One may think of this as tensorization during data acquisition. Incorporating the time variable is a straightforward choice: let an experiment yield matrix data, we can repeat the experiment at various time instances and stack the snapshots in a third-order tensor. Ratings of different users for different movies yield matrix data in movie recommendation systems, while the well-known Netflix dataset additionally includes the rating date yielding a third-order tensor [38]. Other examples can be found, e.g., in social media analysis by taking the time mode into account from e-mail traffic data [7], in food science by collecting flavor release data [287] and in chemometrics by considering the retention time in chromatography [58, 328].

In addition to or besides the temporal diversity, one often exploits the spatial diversity in sensor array processing by providing each receiver with a sensor array rather than a single sensor [323] or in biomedical signal processing by using multiple channels [94, 211]. Experiments can be extended from single-sample or single-subject to multi-sample or multi-subject, e.g., in fluorescence spectroscopy [19, 57], biomedical signal processing [117], face recognition systems [366, 378] and phonetic analysis [183][4]. Tensor faces generalizes standard eigenfaces by incorporating different illuminations, poses and expressions [379]. More generally, one may repeat experiments several times, under varying conditions, with different attributes, etc.

Including additional variables in the measurement process should not be confused with incorporating multiway properties in matrix data. For example, in telecommunications, a unique frequency channel or spreading code is allocated to each source signal. The measured data are strictly speaking still matrix data in function of receiver and time. By model construction, one can acknowledge the multiway character and tensorize the matrix data to a third-order tensor

[4]Note that one of the earliest reported multiway datasets can be found in [183].

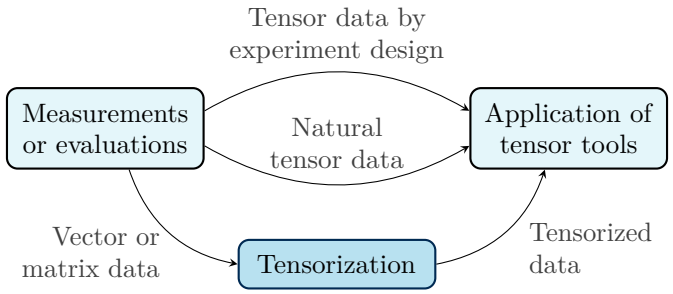


Figure 2.2: Measurements and evaluations from experiments or simulations often readily generate tensor data, which can be processed using various tensor tools. Furthermore, tensor data can be obtained as well by experiment design using additional variables in the data creation process. If only vector/matrix data are available, one can consider an additional tensorization step. The latter concept is the main focus of this paper.

using folding or rearranging (also known as segmentation) with the spectral (resp., spreading) diversity resulting in a third mode [206, 325]. This will be further discussed in Section 2.6.2.

2.4 Functions and derivatives

Various types of multiway data originate from the evaluation of a multivariate function. An evaluation may yield vectors, matrices or tensors of which the order and dimensions depend on the type of function and the type of evaluation used. For example, the evaluation of a tensor-valued function in a single point readily generates a tensor. In Sections 2.4.1 and 2.4.2, we discuss the evaluation of scalar functions, and vector-, matrix- and tensor-valued functions, respectively. In Section 2.4.3, the concept of higher-order derivatives is investigated as a means of increasing the order and, as such, as a tensorization technique. The fundamentals provided in this section form the basis of many tensorization techniques discussed in Sections 2.6 and 2.7.

2.4.1 Evaluation of scalar multivariate functions

We make the distinction between scalar-valued variables in Section 2.4.1 and vector-valued variables in Section 2.4.1.

Scalar functions in scalar-valued variables

Let a tensor $\mathcal{F} \in \mathbb{R}^{100\times100\times100}$ contain temperature data in various locations in a cuboid-shaped room, with f_{ijk} indicating the temperature at spatial coordinates (x_i, y_j, z_k) for $1 \leq i, j, k \leq 100$. It is obvious that the tensor is an evaluated or discretized form of the scalar continuous function $\underline{f}(x, y, z)$.

More generally, let us consider the evaluation of the function $\underline{f}(u^{(1)}, \ldots, u^{(D)})$ in the Cartesian product set $\{x_1^{(1)}, \ldots, x_{I_1}^{(1)}\} \times \cdots \times \{x_1^{(D)}, \ldots, x_{I_D}^{(D)}\}$, yielding the order-$D$ tensor $\mathcal{F} \in \mathbb{K}^{I_1\times I_2\times\cdots\times I_D}$ with entries

$$f_{i_1 i_2 \cdots i_D} = \underline{f}(x_{i_1}^{(1)}, x_{i_2}^{(2)}, \ldots, x_{I_D}^{(D)})$$

for all indices [177]. Some multiway datasets mentioned in Section 2.3 can be seen as originating from multivariate functions. An alternative notation is $\underline{f}(\boldsymbol{u})$ with $\boldsymbol{u} \in \mathbb{K}^D$. We will use both notations throughout this chapter, depending on the context.

Multivariate functions of particular interest are those that are multiplicatively separable in their arguments, i.e., functions that can be written as

$$\underline{f}(u^{(1)}, \ldots, u^{(D)}) = \prod_{d=1}^{D} \underline{g}^{(d)}(u^{(d)}),$$

in which $\underline{g}^{(d)}(u)$ are unknown univariate functions. The function $\underline{f}(u^{(1)}, \ldots, u^{(D)})$ corresponds to a rank-1 tensor $\mathcal{F}$ after evaluation, as $\mathcal{F} = \mathbf{g}^{(1)} \otimes \mathbf{g}^{(2)} \otimes \cdots \otimes \mathbf{g}^{(D)}$ with $g_{i_d}^{(d)} = \underline{g}^{(d)}(x_{i_d}^{(d)})$. An example is the exponential of a linear form:

$$\underline{f}(u^{(1)}, \ldots, u^{(D)}) = \exp(\sum_{d=1}^{D} a_d u^{(d)}) = \prod_{d=1}^{D} \exp(a_d u^{(d)}).$$

Of course, $u^{(d)}$ can be substituted by some univariate function $\underline{h}(u^{(d)})$ such as $(u^{(d)} - b_i)^2$ in the case of Gaussians. The constant a_d then determines the inverse of the variance.

It follows that if a multivariate function $\underline{f}$ can be written as a sum of R separable functions [44, 45], the evaluated tensor $\mathcal{F}$ admits a (C)PD with R rank-1 terms. An example is the function $\underline{f} = \sin(\sum_{d=1}^{D} u^{(d)})$ [44]. Using trigonometric identities, it may seem that $R = 2^{D-1}$ terms are needed. However, it has been proven using multilinear algebra that $R = D$ is sufficient [270].

Scalar functions in vector-valued variables

The scalar function $\underline{f}(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(D)})$ in vector variables $\boldsymbol{u}^{(d)} \in \mathbb{K}^{N_d}$ yields a $I_1 \times \cdots \times I_D$ tensor of order D after evaluation in the Cartesian product set $\{\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_{I_1}^{(1)}\} \times \cdots \times \{\mathbf{x}_1^{(D)}, \ldots, \mathbf{x}_{I_D}^{(D)}\}$ given I_d values for each variable $\mathbf{u}^{(d)}$.

Let us now discuss an interesting variant where the dimensions of $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)}$ are equal, i.e., $N_1, \ldots, N_D = N$. We consider a given set of I data vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I$, and stack the results from evaluating $\underline{f}$ in all possible D-combinations with repetition. An order-D tensor $\mathcal{F} \in \mathbb{K}^{I \times \times \cdots \times I}$ is then obtained with for all indices

$$f_{i_1 i_2 \cdots i_D} = \underline{f}(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_D}).$$

This is equivalent to the evaluation of $\underline{f}$ in the symmetric Cartesian product set $\{\mathbf{x}_1, \ldots, \mathbf{x}_I\} \times \cdots \times \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$. The tensor $\mathcal{F}$ is symmetric if $\underline{f}$ is symmetric in its variables. Given only a single set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I$, we have effectively constructed a tensor $\mathcal{F}$. As we will see, each tensorization technique in Section 2.7 makes use of such a construction.

By way of illustration, a similarity matrix is obtained from a set of objects or data vectors (such as feature vectors) and a similarity function $\underline{f}(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)})$ (such as the cosine or Gaussian similarity functions), by stacking the similarities $\underline{f}(\mathbf{x}_i, \mathbf{x}_j)$ for varying i and j and, hence, by evaluating $\underline{f}$ in the symmetric Cartesian product set. A higher-order similarity tensor then originates from a function $\underline{f}(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(D)})$.

2.4.2 Evaluation of vector-, matrix- and tensor-valued multivariate functions

It is obvious that a tensor function $\underline{\mathcal{F}}(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(D)}) \in \mathbb{K}^{J_1 \times \cdots \times J_E}$ readily yields a tensor $\mathcal{F} \in \mathbb{K}^{J_1 \times \cdots \times J_E}$ when evaluating $\underline{\mathcal{F}}$ in a single point, i.e., a single choice of values for $\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(D)}$. Furthermore, given a set of I vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I$, let us construct the symmetric Cartesian product set $\{\mathbf{x}_1, \ldots, \mathbf{x}_I\} \times \cdots \times \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$ and assume $N_1, \ldots, N_D = N$. Then, one can stack the results from evaluating $\underline{\mathcal{F}}$ in this set in a $J_1 \times \cdots \times J_E \times I \times \cdots \times I$ tensor of order $E + D$.

A first special case is the matrix function $\underline{\mathbf{F}}(\boldsymbol{u}) \in \mathbb{K}^{J_1 \times J_2}$. Such a function corresponds to the mapping of a vector $\mathbf{u}$ to a matrix. Stacking the matrices obtained by evaluating $\underline{\mathbf{F}}(\boldsymbol{u})$ in a given set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I$ yields a third-order tensor $\mathcal{F}^{J_1 \times J_2 \times I}$. Examples will be provided in Section 2.6.

Second, the vector function $\underline{\mathbf{f}}(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}) \in \mathbb{K}^J$ can yield a third-order tensor $\mathcal{F} \in \mathbb{K}^{J \times I \times I}$ as well, after stacking the vectors obtained from evaluating $\underline{\mathbf{f}}$ in the Cartesian product set $\{\mathbf{x}_1, \ldots, \mathbf{x}_I\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$. Alternatively, let us assume the function $\underline{\mathbf{f}}$ can be parametrized by a set of J parameters ρ_j and a scalar function $\underline{f}(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \rho)$ such that $\underline{f}_j(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}) = \underline{f}(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \rho_j)$ for $1 \leq j \leq J$. Evaluating $\underline{f}$ in the Cartesian product set $\{\mathbf{x}_1, \ldots, \mathbf{x}_I\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$ for a particular choice of ρ yields a matrix of size $I \times I$. By stacking the matrices corresponding to the different values $\rho_1, \ldots, \rho_J$, the same third-order tensor $\mathcal{F}$ of size $J \times I \times I$ is obtained. In Section 2.7.1, a number of tensorization techniques related to this type of evaluation are discussed.

Insert 2 — Compound matrix: Let us consider the function $\underline{\mathbf{F}}(\mathbf{u}, \mathbf{v}) = \mathbf{u}\mathbf{v}^{\mathrm{T}} - \mathbf{v}\mathbf{u}^{\mathrm{T}} \in \mathbb{K}^{J \times J}$ with $\mathbf{u}, \mathbf{v} \in \mathbb{K}^J$. Given a set of I vectors $\mathbf{x}_1, \ldots, \mathbf{x}_I \in \mathbb{K}^J$, a fourth-order tensor $\mathcal{F}^{J \times J \times I \times I}$ can be constructed by evaluating $\underline{\mathbf{F}}$ in the Cartesian product set $\{\mathbf{x}_1, \ldots, \mathbf{x}_I\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$. It follows that $\mathbf{F}_{kl} = \mathbf{x}_k \mathbf{x}_l^{\mathrm{T}} - \mathbf{x}_l \mathbf{x}_k^{\mathrm{T}}$ for $1 \leq k, l \leq I$.

This tensor is well known in its reduced unfolded form. Indeed, unfolding $\mathcal{F}$ by combining the first and second mode, and third and fourth mode, respectively, and removing the identical rows and columns yields the compound matrix $\mathbf{C}_2(\mathbf{X}) \in \mathbb{K}^{\frac{J(J-1)}{2} \times \frac{I(I-1)}{2}}$ of the matrix $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1, \ldots, \mathbf{x}_I \end{bmatrix} \in \mathbb{K}^{J \times I}$. The matrix $\mathbf{C}_2(\mathbf{X})$ consists of the determinants of all 2×2 submatrices of $\mathbf{X}$, stacked in lexicographic order. More generally, the Dth compound matrix $\mathbf{C}_D(\mathbf{X})$ consisting of the $D \times D$ minors of $\mathbf{X}$ can be obtained from order-D tensor function generalizations of $\underline{\mathbf{F}}(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)})$.

2.4.3 Tensorization using derivatives

By evaluating a multivariate function in a product set, a tensor is obtained. Derivatives allow an increase of the order of this tensor, without the need of evaluating the function in more points. It follows that derivatives can be well used for tensorization. We make a distinction between scalar and vector functions in Sections 2.4.3 and 2.4.3. Interestingly, derivatives of increasing order can be naturally stacked into a single tensor; this will be discussed below in Section 2.5.2. For simplicity, we limit the discussion in this section to the real case.

Derivatives of multivariate scalar functions

Let us consider again a multivariate scalar function $\underline{f}(u^{(1)}, \ldots, u^{(D)}) \in \mathbb{R}$. The first-order partial derivatives of $\underline{f}$ form the gradient vector $\nabla \underline{f} \in \mathbb{R}^D$ with $(\nabla \underline{f})_d = \partial \underline{f} / \partial u^{(d)}$ whereas the second-order partial derivatives form the (symmetric) Hessian matrix $\Delta \underline{f} \in \mathbb{R}^{D \times D}$. More generally, the partial derivatives of order N can be naturally stacked in a symmetric tensor function $\underline{\mathcal{P}}(u^{(1)}, \ldots, u^{(D)})$ of order N such that

$$\left(\underline{\mathcal{P}}(u^{(1)}, \ldots, u^{(D)})\right)_{i_1 i_2 \ldots i_D} = \frac{\partial^N \underline{f}(u^{(1)}, u^{(2)}, \ldots, u^{(D)})}{\partial u^{(i_1)} \partial u^{(i_2)} \cdots \partial u^{(i_D)}}. \tag{2.3}$$

By evaluating the tensor function in a single point, a tensor $\mathcal{P} \in \mathbb{R}^{D \times D \times \cdots \times D}$ of order N is obtained. Note that the evaluation of a bivariate function $\underline{f}(u, v)$ in a product set yields only a matrix, while the evaluation in a single point of a higher-order derivative of $\underline{f}(u, v)$ yields a $2 \times 2 \times \cdots \times 2$ tensor of order N.

Of particular interest are additively separable scalar functions which can be written as sums of univariate functions. For example, the function $\underline{f}(u^{(1)}, \ldots, u^{(D)})$ can then be written as $\sum_{d=1}^{D} g^{(d)}(u^{(d)})$. It can be seen that the derivatives of order N of such functions are diagonal for $N \geq 2$. The ith diagonal entry of the resulting matrix- or tensor-valued function then equals the derivative $\mathrm{d}^N \underline{g}^{(i)}(u^{(i)}) / \mathrm{d}\left(u^{(i)}\right)^N$. As will be discussed in Section 2.7.3, this diagonality forms the basis of many tensorization-based methods, such as those related to independent component analysis.

Derivatives of multivariate vector functions

Similarly, a multivariate vector function $\underline{\mathbf{f}}(u^{(1)}, \ldots, u^{(D)}) \in \mathbb{R}^J$ (which holds a number of scalar functions) can be tensorized by computing derivatives. The first-order partial derivatives of $\underline{\mathbf{f}}$ form the Jacobian matrix of size $J \times D$ containing the multivariate scalar functions $\partial \underline{f}_j / \partial u^{(d)}$. Evaluating the Jacobian matrix in K points and stacking the obtained matrices along the third mode yields a third-order tensor of dimensions $J \times D \times K$.

More generally, the order-N partial derivatives of $\underline{\mathbf{f}}$ can be collected in a tensor of order $N + 1$ with dimensions $J \times D \times \cdots \times D$ that is symmetric in the last D modes.

It is interesting to consider vector functions of which each scalar function is univariate. Possibly after permuting the variables of $\underline{\mathbf{f}}$, the Jacobian matrix and higher-order tensors consisting of the second- and higher-order partial derivatives are diagonal.

Insert 3 — Decoupling sets of multivariate functions: Consider a set of J multivariate functions $\underline{f}_j(u^{(1)}, \ldots, u^{(D)})$, which together form the vector function $\underline{\mathbf{f}}(u^{(1)}, \ldots, u^{(D)})$. The goal of decoupling the set of multivariate functions, as described in [140], is to obtain a decomposition of the form

$$\underline{f}_j(u^{(1)}, \ldots, u^{(D)}) = \sum_{r=1}^{R} w_{jr} \underline{g}_r \left(\sum_{d=1}^{D} v_{dr} u^{(d)} \right), \tag{2.4}$$

where $\underline{g}_r(z^{(r)})$ are univariate functions in $z^{(r)} = \sum_{d=1}^{D} v_{dr} u^{(d)}$. Such a decoupling can be used to simplify complex nonlinear block-oriented systems [162, 290], among others. If the functions $\underline{f}_j$ and $\underline{g}_r$ are polynomials, the problem becomes a non-homogeneous multi-polynomial variant of the Waring problem discussed in Section 2.5.2 [139].

The unknown matrices $\mathbf{W} \in \mathbb{R}^{J \times R}$ and $\mathbf{V} \in \mathbb{R}^{D \times R}$ can be recovered by evaluating the Jacobian matrix $\underline{\mathbf{J}}$ in K points $(x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(D)})$. A tensor $\mathcal{J} \in \mathbb{R}^{J \times D \times K}$ can be constructed by stacking the obtained matrices along the third mode. Let us denote the derivative of $\underline{g}_r(z^{(r)})$ as $\underline{g}'_r(z^{(r)}) = \mathrm{d}\underline{g}_r(z^{(r)})/\mathrm{d}z^{(r)}$. Based on (2.4), each slice of $\mathcal{J}$ can be written as follows:

$$\mathbf{J}_k = \mathbf{W} \mathrm{diag}\left(\underline{g}'_1(z_k^{(1)}), \ldots, \underline{g}'_R(z_k^{(R)}) \right) \mathbf{V}^{\mathrm{T}},$$

with $z_k^{(r)} = \sum_{d=1}^{D} v_{dr} u_k^{(d)}$. It follows that $\mathcal{T}$ can be written as a sum of the rank-1 terms $\mathbf{w}_r \otimes \mathbf{v}_r \otimes \mathbf{c}_r$ with $c_{kr} = \underline{g}'_r(z_k^{(r)})$ for $1 \leq k \leq K$. If $\mathcal{J}$ admits a unique rank-R CPD, $\mathbf{W}$ and $\mathbf{V}$ can be uniquely recovered. Furthermore, the function values $\underline{g}_r(z^{(r)})$ can be found up to permutation and scaling from (2.4) if $\mathbf{W}$ has full column rank.

2.5 Tensors representing mathematical objects

Rather than modeling multiway data as discussed in the previous sections, tensors can also represent a mathematical object or operator. Indeed, multilinear functions and multivariate polynomials can be represented by tensors, as will be pointed out below. It follows that a number of well-known linear algebra problems and optimization tasks have a clear connection with tensors, such as matrix multiplication and global polynomial optimization.

2.5.1 Relations between multilinear functions and tensors

Multilinear functions are multivariate functions that are linear in each variable. We restrict the discussion to vector variables for convenience.

Multilinear forms

Scalar multilinear functions are also known as multilinear forms. Let us start the discussion with a linear form $\underline{f}$, which maps a vector $\mathbf{u} \in \mathbb{K}^I$ to a scalar. The linearity property implies that $\underline{f}(\alpha_1\mathbf{u}_1 + \alpha_2\mathbf{u}_2) = \alpha_1\underline{f}(\mathbf{u}_1) + \alpha_2\underline{f}(\mathbf{u}_2)$. Writing $\mathbf{u}$ in $\underline{f}(\mathbf{u})$ as $u_1\mathbf{e}_1 + \ldots + u_I\mathbf{e}_I$, $\underline{f}(\mathbf{u})$ can be written as $u_1\underline{f}(\mathbf{e}_1) + \ldots + u_I\underline{f}(\mathbf{e}_I)$. Collecting $\underline{f}(\mathbf{e}_i)$ in $\mathbf{p} \in \mathbb{K}^I$, it follows that $\underline{f}(\mathbf{u}) = \mathbf{u}^{\mathrm{T}}\mathbf{p}$. Hence, each linear form $\underline{f}$ corresponds bijectively to a vector $\mathbf{p}$.

Proceeding further, a bilinear form $\underline{f}$ maps two vectors $\mathbf{u} \in \mathbb{K}^{I_1}$ and $\mathbf{v} \in \mathbb{K}^{I_2}$ to a scalar $\underline{f}(\mathbf{u}, \mathbf{v})$ and is linear in both $\mathbf{u}$ and $\mathbf{v}$. With a similar derivation as before, the bilinear form corresponds to a matrix $\mathbf{P} \in \mathbb{K}^{I_1 \times I_2}$ such that

$$\underline{f}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^{\mathrm{T}}\mathbf{P}\mathbf{v} = \mathbf{P} \cdot_1 \mathbf{u}^{\mathrm{T}} \cdot_2 \mathbf{v}^{\mathrm{T}}, \tag{2.5}$$

in which $\mathbf{P}$ contains the coefficients of the bilinear form when $\mathbf{u}$ and $\mathbf{v}$ are canonical unit vectors. A multilinear form $\underline{f}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ with $\mathbf{u}^{(d)} \in \mathbb{K}^{I_d}$ can then be represented by a tensor $\mathcal{P} \in \mathbb{K}^{I_1 \times \cdots \times I_D}$, and the evaluation of $\underline{f}$ involves some tensor–vector products, as illustrated in Fig. 2.3:

$$\begin{aligned} \underline{f}(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(D)}) &= \mathcal{P} \cdot_1 \boldsymbol{u}^{(1)^{\mathrm{T}}} \cdot_2 \boldsymbol{u}^{(2)^{\mathrm{T}}} \cdots \cdot_D \boldsymbol{u}^{(D)^{\mathrm{T}}} \\ &= \sum_{i_1, i_2, \ldots, i_D} p_{i_1 i_2 \cdots i_D} u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(D)}_{i_D}. \end{aligned} \tag{2.6}$$

A rank-1 tensor $\mathcal{P} = \mathbf{p}^{(1)} \otimes \cdots \otimes \mathbf{p}^{(D)}$ admits a separation of variables:

$$\underline{f}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)}) = \prod_d \left(\mathbf{u}^{(d)^{\mathrm{T}}} \mathbf{p}^{(d)} \right). \tag{2.7}$$

Multilinear vector functions

Let $\underline{\mathbf{f}}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)}) \in \mathbb{K}^J$ denote a multilinear function which maps D variables to a vector of dimension J. This function can be seen as a collection of scalar multilinear functions, each of which can be represented by a tensor of order D.

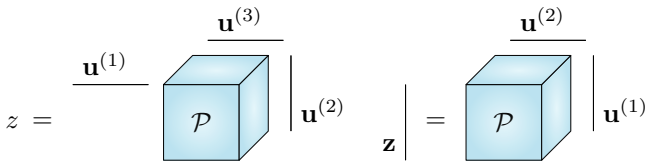


Figure 2.3: Trilinear form $z = \underline{f}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)})$ (left) and bilinear vector function $\mathbf{z} = \underline{\mathbf{f}}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)})$ (right).

Stacking the latter tensors, $\underline{\mathbf{f}}$ can be represented by a tensor $\mathcal{P} \in \mathbb{K}^{J \times I_1 \times \cdots \times I_D}$ of order $D+1$. An evaluation of $\underline{\mathbf{f}}$ then involves tensor–vector products along modes 2 to $D+1$, as illustrated in Fig. 2.3. The reasoning can be easily extended to multilinear matrix and tensor functions.

Examples

A variety of operations in linear algebra are related to multilinear forms and functions. The trace of a matrix corresponds to a linear form, while the inner product of two vectors is a trivial bilinear form with $\mathbf{P} = \mathbf{I}$. The matrix determinant is linear in each of the columns of the matrix, and hence corresponds to a multilinear form. As a special case, the determinant of a 2×2 matrix is the bilinear mapping represented by

$$\mathbf{P} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

The matrix multiplication of two matrices and the multiplication of complex numbers both correspond to a bilinear function, as well as the convolution of two vectors.

Insert 4 — Matrix multiplication: Let us consider the multiplication of two square $I \times I$ matrices $\mathbf{A}$ and $\mathbf{B}$, yielding $\mathbf{C} = \mathbf{AB}$. The product maps $\mathbf{A}$ and $\mathbf{B}$ in a particular way to $\mathbf{C}$. This can be written as $\text{vec}(\mathbf{C}) = \underline{f}(\text{vec}(\mathbf{A}), \text{vec}(\mathbf{B}))$ with the function $\underline{f}$ denoting the multiplication operation. It is a bilinear transformation, as $(\alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2)\mathbf{B} = \alpha_1(\mathbf{A}_1\mathbf{B}) + \alpha_2(\mathbf{A}_2\mathbf{B})$ and $\mathbf{A}(\beta_1\mathbf{B}_1 + \beta_2\mathbf{B}_2) = \beta_1\mathbf{A}\mathbf{B}_1 + \beta_2\mathbf{A}\mathbf{B}_2$. As discussed, $\underline{f}$ corresponds to a tensor $\mathcal{P} \in \mathbb{K}^{I^2 \times I^2 \times I^2}$, such that $\text{vec}(\mathbf{C}) = \mathcal{P} \cdot_2 \text{vec}(\mathbf{A}) \cdot_3 \text{vec}(\mathbf{B})$. Such a tensor $\mathcal{P}$ is called a Strassen multiplication tensor [350]. Computing $\mathbf{C}$ using a polyadic decomposition of $\mathcal{P}$ has led to faster algorithms than the standard approach. The computational complexity depends on the (border)

rank of $\mathcal{P}$. In the case of 2×2 matrices, for example, the number of scalar multiplications can be reduced from 8 to 7, the latter being the rank of the corresponding $4 \times 4 \times 4$ multiplication tensor [234, 350].

2.5.2 Relations between polynomials and tensors

Studies of homogeneous[5] polynomials (also called quantics) in the 19th century form the roots of multiway analysis [75, 188]. Homogeneous polynomials can be seen as symmetric versions of multilinear forms, as will be discussed in Section 2.5.2. Connections between tensors and non-homogeneous polynomials follow in Section 2.5.2. For a profound discussion and further references we refer to [233].

Homogeneous multivariate polynomials

Let $\mathbf{A}$ be a symmetric matrix of size $I \times I$. The product of $\mathbf{A}$ with a vector $\mathbf{u} \in \mathbb{K}^I$ along both modes yields a quadratic form:

$$\underline{p}(\boldsymbol{u}) = \boldsymbol{u}^{\mathrm{T}} \mathbf{A} \boldsymbol{u} = \mathbf{A} \bullet_1 \boldsymbol{u}^{\mathrm{T}} \bullet_2 \boldsymbol{u}^{\mathrm{T}} \tag{2.8}$$

$$= \sum_{d=1}^{D} a_{dd} u_d^2 + \sum_{d_1 < d_2}^{D} 2 a_{d_1 d_2} u_{d_1} u_{d_2}.$$

It can be seen that $\underline{p}(\mathbf{u})$ is an homogeneous polynomial of degree 2 in I variables. More generally, an homogeneous polynomial $\underline{p}(\mathbf{u})$ of degree D can be associated with a symmetric tensor $\mathcal{A} \in \mathbb{K}^{I \times \cdots \times I}$ of order D:

$$\underline{p}(\boldsymbol{u}) = \mathcal{A} \bullet_1 \boldsymbol{u}^{\mathrm{T}} \bullet_2 \boldsymbol{u}^{\mathrm{T}} \cdots \bullet_D \boldsymbol{u}^{\mathrm{T}}. \tag{2.9}$$

It can be seen that (2.8) and (2.9) are symmetric variants of (2.5) and (2.6), respectively. Note that $\underline{p}(\mathbf{u})$ is not linear in $\mathbf{u}$.

Insert 5 — The polynomial Waring problem: If $\mathcal{A}$ in (2.9) is a rank-1 tensor, i.e., $\mathcal{A} = \mathbf{b} \otimes \cdots \otimes \mathbf{b}$, the polynomial $\underline{p}(\mathbf{u})$ corresponds to the Dth power of a linear form:

$$\underline{p}(\boldsymbol{u}) = (\mathbf{b} \otimes \cdots \otimes \mathbf{b}) \bullet_1 \boldsymbol{u}^{\mathrm{T}} \cdots \bullet_D \boldsymbol{u}^{\mathrm{T}} = (\boldsymbol{u}^{\mathrm{T}} \mathbf{b})^D .$$

[5] Recall that an homogeneous polynomial consists of terms of the same degree.

It follows that if $\mathcal{A}$ admits a rank-R CPD, we can write $\underline{p}(\boldsymbol{u}) = \sum_{r=1}^{R} (\boldsymbol{u}^{\mathrm{T}}\mathbf{b}_r)^D$. Finding such a minimal sum of R Dth-power linear forms given a polynomial $\underline{p}(\mathbf{u})$ is known as the (polynomial) Waring problem [281].

In the quadratic case, the problem can be solved using diagonalization by congruence, using Lagrange's multiplier method or using an eigenvalue decomposition of $\mathbf{A}$ [26, 272]. The general Waring problem for $D > 2$, on the other hand, is more difficult. It is equivalent with finding the symmetric (C)PD $\mathcal{A} = [\![\mathbf{B}, \ldots, \mathbf{B}]\!]$ with $\mathbf{B} = \left[\mathbf{b}_1, \ldots, \mathbf{b}_R\right]$. The minimal number of terms in the Waring decomposition is equal to the symmetric rank of $\mathcal{A}$ [91, 92].

Insert 6 — From linear to polynomial classification: In binary linear classification, a vector $\mathbf{a} \in \mathbb{K}^I$ is trained or designed such that a given vector $\mathbf{f}$ can be classified according to the sign of $\mathbf{f}^{\mathrm{T}}\mathbf{a}$. Two classes in the I-dimensional space are then separated by a vector hyperplane. The linear function can be generalized to an homogeneous polynomial of, for example, degree 3. The sign of $\underline{p}(\mathbf{f})$ with $\underline{p}(\boldsymbol{u}) = \mathcal{A} \cdot_1 \boldsymbol{u}^{\mathrm{T}} \cdot_2 \boldsymbol{u}^{\mathrm{T}} \cdot_3 \boldsymbol{u}^{\mathrm{T}} = (\boldsymbol{u} \otimes \boldsymbol{u} \otimes \boldsymbol{u})^{\mathrm{T}} \mathrm{vec}\,(\mathcal{A})$ then provides a basis for classification. As described in the next section, the homogenization trick allows the extension to affine hyperplanes and non-homogeneous polynomials. By constraining $\mathcal{A}$ to a tensor with low (multilinear) rank, a reduction in the number of variables can be obtained. This dimension reduction can improve the interpretability and prevent overfitting [304].

Insert 7 — Homogeneous polynomial optimization on the unit sphere: Consider a homogeneous polynomial $\underline{g}(\mathbf{v})$ of degree D in $\mathbf{v} \in \mathbb{R}^I$ that is associated with a symmetric tensor $\mathcal{G} \in \mathbb{R}^{I \times \cdots \times I}$ of order D so that $\underline{g}(\mathbf{v}) = \mathcal{G} \cdot_1 \mathbf{v}^{\mathrm{T}} \cdots \cdot_D \mathbf{v}^{\mathrm{T}}$. Finding the global extremum of $\underline{g}(\mathbf{v})$ on the unit sphere (i.e., with maximum absolute value $\left|\underline{g}(\mathbf{v})\right|$ for $||\mathbf{v}|| = 1$) is equivalent to the problem of maximizing $(\underline{g}(\mathbf{v}))^2$ which in its turn is equivalent to the minimization of $||\mathcal{G} - [\![\lambda;\, \mathbf{v}, \ldots, \mathbf{v}]\!]||^2$ [112]. Hence, the original problem corresponds to finding the best (symmetric) rank-1 approximation of $\mathcal{G}$. The scalar λ and vector $\mathbf{v}$ are also known as the largest Z-eigenvalue and corresponding Z-eigenvector of $\mathcal{G}$, respectively [222, 301–303].

Non-homogeneous multivariate polynomials

A non-homogeneous multivariate polynomial can be characterized by two types of degrees: its total degree and its coordinate degree. The total degree denotes the maximum of the degrees of the monomials, while the coordinate degree is a tuple consisting of the degrees of the univariate polynomials in each of the variables. For example, a bivariate polynomial with total degree 2 can contain the terms u^2, uv and v^2, while a coordinate degree of $(1, 1)$ allows the term uv but not the terms u^2 and v^2. Each type has its own connection with tensors.

Let us first consider a non-homogeneous polynomial $\underline{p}(\mathbf{u})$ of total degree D in $\mathbf{u} \in \mathbb{K}^I$ which can be expressed as

$$\underline{p}(\boldsymbol{u}) = a^{(0)} + \boldsymbol{u}^{\mathrm{T}}\mathbf{a}^{(1)} + \boldsymbol{u}^{\mathrm{T}}\mathbf{A}^{(2)}\boldsymbol{u} + \cdots$$

$$+ \mathcal{A}^{(D)} \bullet_1 \boldsymbol{u}^{\mathrm{T}} \cdots \bullet_D \boldsymbol{u}^{\mathrm{T}},$$

with constant term $a^{(0)} \in \mathbb{K}$ and with $\mathbf{a}^{(1)} \in \mathbb{K}^I, \ldots, \mathcal{A}^{(D)} \in \mathbb{K}^{I \times \cdots \times I}$ holding the coefficients of the monomials. Let us symmetrically stack the coefficients in a larger order-D tensor $\mathcal{A}^{(\mathrm{hom})} \in \mathbb{K}^{(I+1)\times\cdots\times(I+1)}$, as shown in Fig. 2.4. The non-homogeneous polynomial can then also be written as

$$\underline{p}(\boldsymbol{u}) = \mathcal{A}^{(\mathrm{hom})} \bullet_1 \left[1, \boldsymbol{u}^{\mathrm{T}}\right] \cdots \bullet_D \left[1, \boldsymbol{u}^{\mathrm{T}}\right].$$

This process is called homogenization, and the homogeneous polynomial $\underline{p}^{(\mathrm{hom})}(\boldsymbol{v}) = \mathcal{A}^{(\mathrm{hom})} \bullet_1 \boldsymbol{v}^{\mathrm{T}} \cdots \bullet_D \boldsymbol{v}^{\mathrm{T}}$ is the homogenized form of $\underline{p}(\boldsymbol{u})$, with $\underline{p}^{(\mathrm{hom})}(\left[1, \boldsymbol{u}^{\mathrm{T}}\right]^{\mathrm{T}}) = \underline{p}(\boldsymbol{u})$. Hence, a symmetric tensor of size $(I+1) \times \cdots \times (I+1)$ cannot only represent an homogeneous polynomial in $I+1$ variables, as discussed in Section 2.5.2, but also a non-homogeneous polynomial in I variables. Using homogenization, the three inserts from Section 2.5.2 can be extended to non-homogeneous polynomials.

Let us now consider a non-homogeneous polynomial $\underline{p}(\mathbf{u})$ of coordinate degree $(D_1, \ldots, D_I)$ with $\mathbf{u} \in \mathbb{K}^I$. Unlike in the previous paragraph, $\underline{p}$ will not be represented by a symmetric tensor $\mathcal{A}$ of size $(I+1)\times\cdots\times(I+1)$, but rather by a possibly unsymmetric tensor $\mathcal{C}$ of size $(D_1+1)\times\cdots\times(D_I+1)$. Recall that $\underline{p}(\mathbf{u})$ is a polynomial of degree D_i in the ith variable, and define the Vandermonde vector $\mathbf{v}(u_i) = \left[1, u_i, u_i^2, \ldots, u_i^{D_i}\right]^{\mathrm{T}}$. It follows that

$$\underline{p}(\boldsymbol{u}) = \mathcal{C} \bullet_1 \mathbf{v}(u_1)^{\mathrm{T}} \cdots \bullet_I \mathbf{v}(u_I)^{\mathrm{T}}.$$

Observe the connection with the multilinear form of (2.6), which is now evaluated in Vandermonde vectors. Also, if $\mathcal{C}$ has rank 1 then $\underline{p}(\mathbf{u})$ is a product of univariate polynomials, similar to the separation of variables in (2.7).

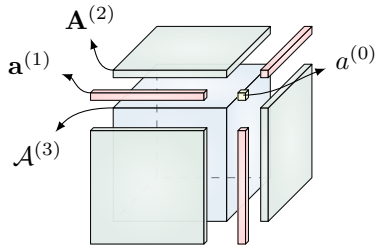


Figure 2.4: Illustration of the natural stacking of the coefficient tensors of a cubic (degree-3) multivariate non-homogeneous polynomial with I variables in a tensor $\mathcal{A}^{(\text{hom})}$ with dimensions $(I+1) \times (I+1) \times (I+1)$, which corresponds to the coefficient tensor of a cubic homogenized polynomial with $I+1$ variables. Note that the tensor has been rotated for illustration purposes as $a^{(0)} = a_{111}$.

Insert 8 — Non-homogeneous polynomial optimization over a finite interval: Let $\underline{f}(u)$ be a non-homogeneous polynomial of degree D. We want to find the global extremum of $\underline{f}(u)$ over a finite interval. Without loss of generality, we consider $u \in [-1, 1]$. By a change of variables[6], the function $\underline{f}(u)$ can be transformed to a homogeneous polynomial $\underline{g}(\mathbf{v}) = \mathcal{G} \bullet_1 \mathbf{v}^{\mathrm{T}} \cdots \bullet_{2D} \mathbf{v}^{\mathrm{T}}$ with the unit-norm vector $\mathbf{v} \in \mathbb{R}^2$ depending on u. The transformation maps the finite interval $[-1, 1]$ to the unit circle. The entries in $\mathcal{G}$ depend only on the coefficients of $\underline{f}$. Recalling Insert 7, finding the global finite-interval extremum of $\underline{f}(u)$ is equivalent to finding a best symmetric rank-1 tensor approximation.

Insert 9 — From systems of linear equations to systems of polynomial equations: A system of linear equations is commonly represented by a matrix equation $\mathbf{Au} = \mathbf{b}$. Each row of $\mathbf{A}$ corresponds to a linear form in $\mathbf{u}$. Systems of polynomial equations are interesting generalizations. They are structurally richer than linear systems while easier to work with than general nonlinear systems. The following is an example of a bivariate system

[6]More specifically, we may parametrize u as $u = \cos t = \cos^2 \frac{t}{2} - \sin^2 \frac{t}{2}$ with $t \in [0, \pi]$. Multiplying each degree-d monomial of $\underline{f}(u)$ by the trivial factor $1 = \left(\cos^2 \frac{t}{2} + \sin^2 \frac{t}{2}\right)^{D-d}$ yields a degree-$2D$ polynomial that is homogeneous in the entries of $\mathbf{v} = \left[\cos \frac{t}{2}, \sin \frac{t}{2}\right]^{\mathrm{T}}$. Indeed, $||\mathbf{v}|| = 1$.

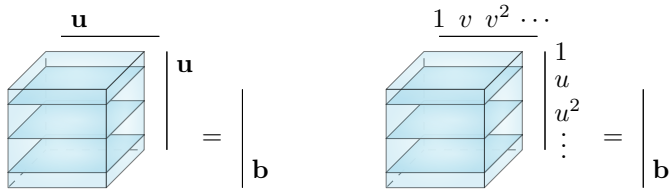


Figure 2.5: Systems of polynomial equations are represented in two ways: based on the polynomials' total degree (left, illustrated for multivariate polynomials with total degree two) and based on the polynomials' coordinate degree (right, illustrated for bivariate polynomials).

of polynomials of low degree:

$$\begin{cases} 3u^2 + v - uv = 3 \\ v^3 - u + u^2 v = 5 \end{cases}$$

As discussed, each polynomial equation can be uniquely represented by a coefficient matrix or tensor. As illustrated in Fig. 2.5, the global polynomial system can be represented by a single tensor by stacking the individual coefficient matrices or tensors. While a system of linear equations provides a more local representation of a phenomenon, a system of polynomial equations can yield an accurate representation in a larger area of operation. Fig. 2.6 shows the solution(s) of a linear system and of a polynomial system of moderate degree. Besides Gröbner approaches, homotopy continuation techniques and methods based on the Macaulay null space or polynomial eigenvalue problem [29, 61, 95, 179, 243, 336], a polynomial system can also be solved by computing a CPD [336, 375].

Linearization and series expansion

Modeling a non-linear multivariate function $\underline{f}(\mathbf{u})$ using a low number of parameters is a common problem within the domains of optimization theory, network analysis and systems theory, among others. Linearization, for example, yields a linear function which provides a good approximation in a small region around a provided operating point $\mathbf{u}_0$:

$$\underline{f}(\mathbf{u}) \approx \underline{f}(\mathbf{u}_0) + \Delta\mathbf{u}^{\mathrm{T}}\mathbf{a},$$

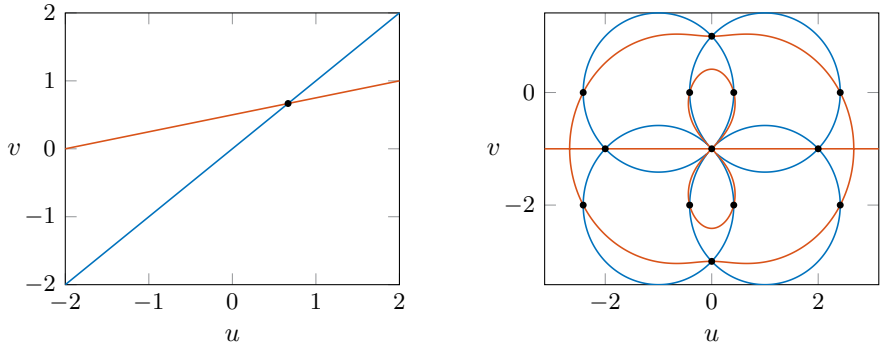


Figure 2.6: An example of the solution(s) of a bivariate linear system (left) and of a bivariate polynomial system (right, taken from [336]), illustrating that polynomial systems are significantly richer than linear systems.

with $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}_0$ and with $\mathbf{a} = \nabla \underline{f} \in \mathbb{K}^I$ the gradient of $\underline{f}$ evaluated in $\mathbf{u}_0$. Instead of approximating $\underline{f}$ by a linear function, a polynomial can be used. The latter can provide a better approximation of $\underline{f}$ in a larger region around $\mathbf{u}_0$, as illustrated in Fig. 2.7. For example, a Taylor series expansion of degree D then yields:

$$\underline{f}(\mathbf{u}) \approx \underline{f}(\mathbf{u}_0) + \Delta\mathbf{u}^{\mathrm{T}}\mathbf{a}^{(1)} + \Delta\mathbf{u}^{\mathrm{T}}\mathbf{A}^{(2)}\Delta\mathbf{u}$$
$$+ \ldots + \mathcal{A}^{(D)} \bullet_1 \Delta\mathbf{u}^{\mathrm{T}} \ldots \bullet_D \Delta\mathbf{u}^{\mathrm{T}},$$

in which $\mathcal{A}^{(d)} = \frac{1}{d!}\mathcal{P}^{(d)}$ with $\mathcal{P}^{(d)}$ consisting of the derivatives of order d of $\underline{f}$ evaluated in $\mathbf{u}_0$. This series of higher-order derivatives can be collected in a single tensor as discussed in Section 2.4.3 and as illustrated in Fig. 2.4. It follows that a multivariate non-linear function can be well represented by a single tensor using polynomial approximation.

2.6 Tensorization of a single vector

The availability of powerful tensor techniques has allowed successful results in various applications which can not be obtained using linear algebra in a straightforward way. A number of examples have been provided in Section 2.3 if tensor data are readily available from measurements, observations or evaluations. However, given only vector or matrix data, tensor tools can still be applied by first tensorizing the data to a tensor.

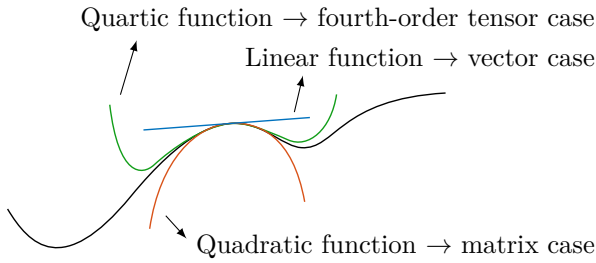


Figure 2.7: Illustration of the approximation of a non-linear function using a linear function, a quadratic function and a quartic function. These approximations of multivariate functions can be represented by vectors, matrices and fourth-order tensors, respectively. Quartic functions provide a better approximation in a broader range compared to linear functions.

In this section, we will discuss tensorization techniques given a single vector. This vector can be a time series or an evaluated univariate function, among others. Note that the techniques can also be applied on a matrix by tensorizing each column and by stacking the results. The tensorization step translates the assumed vector model to a tensor model, meaning that the obtained tensor satisfies specific properties for specific types of vectors. We will uncover such relations. Section 2.7 then presents techniques that tensorize a given matrix and that are not suited in general to be applied on a single vector. Although to a much lesser extent, some techniques have appeared in earlier surveys [118, 296].

More mathematically, different matrix-valued functions $\underline{\mathbf{Z}}(\mathbf{f})$ are presented which map or transform a data vector $\mathbf{f}$ to a matrix $\mathbf{Z}$. Furthermore, we present meaningful mappings of a given vector $\mathbf{f}$ to a tensor $\mathcal{Z}$ using tensor-valued functions $\underline{\mathcal{Z}}(\mathbf{f})$. For each method, links are established between the properties of $\mathbf{f}$ and the properties of $\mathbf{Z}$ or $\mathcal{Z}$.

2.6.1 Hankel/Toeplitz matrices and tensors

We discuss the strong relationship between Hankel/Toeplitz matrices/tensors on the one hand, and polynomials and exponential functions on the other hand, the latter having a broad relevance for applications such as (multidimensional) harmonic retrieval and direction-of-arrival estimation [207, 309, 322, 323]. We limit the discussion to Hankel matrices/tensors as Toeplitz matrices/tensors are permuted versions; the results can be transferred directly. Unlike Toeplitz matrices/tensors, Hankel matrices/tensors offer symmetry in the expressions.

Informally, we call the process of constructing such Hankel matrices/tensors *Hankelization*.

Rank-1 and low-rank Hankel matrices from exponentials, sinusoids and polynomials

Given a data vector $\mathbf{f} \in \mathbb{K}^N$, a Hankel matrix $\mathbf{H} \in \mathbb{K}^{I \times J}$ with $N = I + J - 1$ is defined as

$$\mathbf{H} = \begin{bmatrix} f_1 & f_2 & f_3 & \cdots & f_J \\ f_2 & f_3 & f_4 & \cdots & f_{J+1} \\ f_3 & f_4 & f_5 & \cdots & f_{J+2} \\ \vdots & \vdots & \vdots & & \vdots \\ f_I & f_{I+1} & f_{I+2} & \cdots & f_N \end{bmatrix}, \tag{2.10}$$

with the entries $h_{ij} = f_{i+j-1}$ for $1 \leq i \leq I$ and $1 \leq j \leq J$. It seems that Hankel matrices can have low rank via two different types of functions: exponentials and polynomials.

First, it is well known that $\mathbf{H}$ has rank 1 if and only if $\mathbf{f}$ is an exponential $\underline{f}(t) = cz^t$ evaluated in an equidistant point set [231]. Given[7] $\mathbf{f} = \begin{bmatrix} c, cz, cz^2, \ldots, cz^{N-1} \end{bmatrix}^{\mathrm{T}}$, we obtain the following decomposition with exponential factor vectors $\mathbf{a}$ and $\mathbf{b}$:

$$\mathbf{H} = c\,\mathbf{a}\mathbf{b}^{\mathrm{T}} \quad \text{with} \quad \begin{cases} \mathbf{a} = \begin{bmatrix} 1, z, z^2, \ldots, z^{I-1} \end{bmatrix}^{\mathrm{T}} \in \mathbb{K}^I, \\ \mathbf{b} = \begin{bmatrix} 1, z, z^2, \ldots, z^{J-1} \end{bmatrix}^{\mathrm{T}} \in \mathbb{K}^J. \end{cases} \tag{2.11}$$

This result can be readily generalized to sums-of-exponentials. While a single exponential leads to a rank-1 Hankel matrix, a sum of R exponentials with different poles z_r yields a rank-R Hankel matrix $\mathbf{H}$. The latter admits the so-called Vandermonde decomposition with Vandermonde factor matrices $\mathbf{A}$ and $\mathbf{B}$ [376]. Sinusoids and exponentially-damped sinusoids are special cases of sums-of-exponentials as, through Euler's formula, they can be written as a sum of two exponentials. For example, one can write $e^{at}\cos(\omega t)$ as $\frac{1}{2}e^{(j\omega+a)t} + \frac{1}{2}e^{(-j\omega+a)t}$.

Second, regarding polynomials, a constant function c leads to a constant rank-1 Hankel matrix[8], while a linear function $\underline{f}(t) = ct$ evaluated in $\{0, 1, \ldots, N-1\}$ leads to a rank-2 Hankel matrix. Indeed, it can be seen that $h_{ij} = c(i + j) =$

[7] We sample in the points $\{0, 1, \ldots, N-1\}$; nevertheless, the results apply for any set of equidistant points.

[8] Note that a constant function is actually an exponential with $z = 1$.

Table 2.1: Ranks of the Hankel matrices constructed from different types of evaluated exponential polynomials $\underline{f}(t)$ [51], with $\underline{p}_r(t)$ a polynomial of degree Q_r. The dimensions of $\mathbf{H}$ are assumed sufficiently large.

$f(t)$	$r(\mathbf{H})$	$f(t)$	$r(\mathbf{H})$
cz^t	1	$\sum_{r=1}^{R} c_r z_r^t$	R
$c\sin(\omega t+\phi)$ $c\cos(\omega t+\phi)$	2	$\sum_{r=1}^{R} c_r \sin(\omega_r t)$	$2R$
$cz^t \sin(\omega t)$	2	$\sum_{r=1}^{R} c_r z_r^t \sin(\omega_r t)$	$2R$
$p(t)=\sum_{q=0}^{Q} c_q t^q$	$Q+1$	$\sum_{r=1}^{R} p_r(t)$	$\sum_{r=1}^{R} Q_r + R$
$p(t)z^t$	$Q+1$	$\sum_{r=1}^{R} p_r(t) z_r^t$	$\sum_{r=1}^{R} Q_r + R$
$p(t)\sin(\omega t)$	$2Q+2$	$\sum_{r=1}^{R} p_r(t)\sin(\omega_r t)$	$\sum_{r=1}^{R} Q_r + 2R$
$p(t)z^t \sin(\omega t)$	$2Q+2$	$\sum_{r=1}^{R} p_r(t) z_r^t \sin(\omega_r t)$	$\sum_{r=1}^{R} Q_r + 2R$

$ci + cj$ for all indices and hence

$$\mathbf{H} = c \begin{bmatrix} 0 \\ 1 \\ \vdots \\ I-1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + c \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \\ \vdots \\ J-1 \end{bmatrix}.$$

Recall that $\otimes$ denotes the outer product $\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^{\mathrm{T}}$. It follows that a general polynomial of degree Q yields a Hankel matrix of rank $Q+1$.

These results form the basis of the set of low-rank Hankel properties summarized in Table 2.1. Exponential polynomials, which are sums and/or products of exponentials, sinusoids and/or polynomials, form the general class of functions leading to low-rank Hankel matrices [101]. They can be seen as sums-of-exponentials with poles possibly coinciding in the limit, and can be used to model and approximate a variety of shapes. The well-known (truncated) Fourier series expansion, for example, is a specific case of such an approximation.

Hankel tensors

The results can be extended to higher orders in a straightforward manner. By Hankelizing each column (or row) of a Hankel matrix multiple times, a higher-order Hankel tensor is obtained. In essence, given a data vector $\mathbf{f} \in \mathbb{K}^N$, we obtain a Hankel tensor $\mathcal{H} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_D}$ with entries $h_{i_1 i_2 \ldots i_D} = f_{i_1+i_2+\cdots+i_D-D+1}$ for all indices and with $N = (\sum_d I_d) - D + 1$ [294]. While Hankel matrices have constant anti-diagonals, Hankel tensors have constant anti-diagonal hyperplanes. Note that Hilbert tensors [330] and anti-circulant tensors [129] are specific cases of Hankel tensors.

An exponential $\mathbf{f} = \begin{bmatrix} c, cz, \ldots, cz^{N-1} \end{bmatrix}^{\mathrm{T}}$ leads to a rank-1 Hankel tensor with Vandermonde factor vectors:

$$\mathcal{H} = c \begin{bmatrix} 1 \\ z \\ z^2 \\ \vdots \\ z^{I_1-1} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \\ z^2 \\ \vdots \\ z^{I_2-1} \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 \\ z \\ z^2 \\ \vdots \\ z^{I_D-1} \end{bmatrix} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_D}.$$

More generally, a Hankel tensor constructed from a sum of R exponentials with distinct poles (such as a sinusoid) has rank R. It admits the CPD $\left[\!\left[\mathbf{c}; \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(D)} \right]\!\right]$ with Vandermonde matrices $\mathbf{U}^{(d)}$, which is a direct generalization of the Vandermonde decomposition in the matrix case.

The Hankel tensor $\mathcal{H}$ constructed from a degree-Q polynomial, on the other hand, has a more sophisticated structure. It appears that $\mathcal{H}$ has multilinear rank $(Q+1, \ldots, Q+1)$ and admits a LMLRA with a highly structured core tensor [101, 149, 376]. Consequently, splitting a given function or signal into exponential polynomial components involves a BTD rather than a CPD [101].

Insert 10 — Exponentials and quantization: Let $\mathbf{f} = [1, z, \ldots, z^{N-1}]$ denote an exponential with N a power of 2 such that $D = \log_2 N$. The exponential $\mathbf{f}$ can be represented by a rank-1 Hankel matrix/tensor. Let us apply $D-1$ Hankelization steps, resulting in a $2 \times 2 \times \cdots \times 2$ Hankel tensor $\mathcal{H}$ of order 2. This tensor admits the following symmetric 'quantized' representation [215]:

$$\mathcal{H} = \begin{bmatrix} 1 \\ z \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 \\ z \end{bmatrix}.$$

Insert 11 — Pole estimation and harmonic retrieval: Let us consider the estimation of the poles of a sum-of-exponential, originally solved using Prony's method [284, 365], or using ESPRIT [309] and related matrix pencil-based methods [194]. This fundamental problem appears in many forms, e.g., in the shape-from-moments problem where the poles indicate the vertices of a to-be-recovered polygon [145], or in signal processing where the poles indicate the frequencies of harmonics [231, 293, 294]. Assuming a model with R poles and provided that enough samples are available, the constructed Hankel matrix/tensor has rank R. Two basic approaches are available to solve the problem.

On the one hand, the poles can be found by exploiting the shift invariance in the row, column or mode-n fiber space of the Hankel matrix/tensor. The matrix case leads to ESPRIT while subspace algorithms are available in the tensor case as well [176, 307]. These algorithms are mostly based on the (ML)SVD, (G)EVD, and/or LMLRA.

On the other hand, one can estimate the poles by extracting the rank-1 terms and further neglecting the exponential structure. Although such a factorization in the matrix case does not yield unique rank-1 terms, the CPD in the tensor case is unique under mild conditions on the poles [323, 337, 340, 341]. Note that unlike in the first approach, R can exceed the dimensions of the Hankel tensor in this CPD-based approach.

Efficient tensor decompositions

Because of the repetition of entries, the size of the Hankel tensor can significantly exceed the original number of data points to the point of becoming unmanageable. However, we can avoid the construction of the tensor and circumvent the so-called curse of dimensionality by exploiting the Hankel structure in the tensor decomposition algorithm [389–391]. For instance, a tensor–vector multiplication involving a Hankel tensor of size $I_1 \times I_2 \times \cdots \times I_D$ requires $\mathcal{O}(\prod_{d=1}^{D} I_d)$ flops with a naive implementation. Using fast Fourier transforms, the complexity can be reduced to $\mathcal{O}(N \log N)$ flops with $N = \sum_{d=1}^{D} I_d$ [27, 129]. The memory cost consists of only storing the original N data values instead of storing all $\prod_{d=1}^{D} I_d$ entries.

2.6.2 Segmentation, decimation, folding and reshaping

A data vector can be reshaped or folded to a matrix in two different ways. While segmentation stacks consecutive parts of the vector, decimation stacks downsampled versions.

Given a data vector $\mathbf{f} \in \mathbb{K}^N$, we extract J consecutive segments of length I with $N = IJ$ and stack the segments as the columns of a matrix $\mathbf{S} \in \mathbb{K}^{I \times J}$:

$$\mathbf{S} = \begin{bmatrix} f_1 & f_{I+1} & \cdots & f_{(J-1)I+1} \\ f_2 & f_{I+2} & \cdots & f_{(J-1)I+2} \\ \vdots & \vdots & & \vdots \\ f_I & f_{2I} & \cdots & f_N \end{bmatrix}. \tag{2.12}$$

For instance, a climate temperature time series can be segmented to a matrix, each row and column corresponding to a particular day and year, respectively [413].

Alternatively, one can decimate $\mathbf{f}$ with a certain subsampling factor I and stack the J decimated vectors, resulting in a matrix $\mathbf{D} \in \mathbb{K}^{J \times I}$:

$$\mathbf{D} = \begin{bmatrix} f_1 & f_2 & \cdots & f_I \\ f_{I+1} & f_{I+2} & \cdots & f_{I+1} \\ f_{2I+1} & f_{2I+2} & \cdots & f_{I+2} \\ \vdots & \vdots & & \vdots \\ f_{(J-1)I+1} & f_{(J-1)I+2} & \cdots & f_N \end{bmatrix}.$$

As $\mathbf{D}$ is the transpose of $\mathbf{S}$ in (2.12), we will only discuss low-rank properties of (variants of) segmentation.

Rank-1 matrices

Let us consider a periodic function with period equal to (a multiple of) I. It is straightforward to see that a corresponding matrix $\mathbf{S}$ has rank 1. An exponential leads to a rank-1 matrix as well. Third, a DS-CDMA signal also leads to a rank-1 matrix if I equals (a multiple of) the spreading code length [325].

More generally, it can be seen that $\mathbf{S}$ has rank 1 and can be written as $\mathbf{S} = \mathbf{a}\mathbf{b}^{\mathrm{T}}$ if and only if $\mathbf{f}$ can be written as the Kronecker product $\mathbf{f} = \mathbf{b} \otimes \mathbf{a}$. A periodic signal then results in the vector $\mathbf{a}$ containing one or multiple signal portions of period length and in a constant vector $\mathbf{b}$. An exponential leads to Vandermonde vectors $\mathbf{a}$ and $\mathbf{b}$ while in a DS-CDMA signal the vectors $\mathbf{a}$ and $\mathbf{b}$ consist of the original sequence of symbols and the spreading code, respectively.

Rank-R matrices and connections with Hankelization

A close link between Hankelization and segmentation emerges. Given a vector $\mathbf{f} \in \mathbb{K}^N$, let us consider the matrices $\mathbf{S} \in \mathbb{K}^{I \times J}$ and $\mathbf{H} \in \mathbb{K}^{I \times (N-I+1)}$ obtained from segmentation and Hankelization, respectively, with $N = IJ$. The columns of $\mathbf{S}$ form a subset of the columns of $\mathbf{H}$ so that $r(\mathbf{S}) \leq r(\mathbf{H})$ [51]. The latter inequality can be understood from the fact that $\mathbf{S}$ offers a more compact representation than $\mathbf{H}$. It follows that a vector $\mathbf{f}$ leading to a low-rank matrix $\mathbf{H}$ also results in a low-rank matrix $\mathbf{S}$.

More specifically, the matrix $\mathbf{S} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$ has rank R if and only if $\mathbf{f}$ can be written as the sum of R Kronecker products, with $\mathbf{f} = \sum_{r=1}^{R} \mathbf{b}_r \otimes \mathbf{a}_r$ and full column rank matrices $\mathbf{A} \in \mathbb{K}^{I \times R}$, $\mathbf{B} \in \mathbb{K}^{J \times R}$. Referring to the examples in Section 2.6.2, a low-rank matrix rather than a rank-1 matrix may be obtained if the segment length I is not exactly equal to the period of the periodic function (e.g., if the period is not exactly known). Second, sums-of-exponentials, polynomials and exponential polynomials all give rise to a low-rank matrix $\mathbf{S}$ depending on their degree [51]. Finally, a convolution involving for example DS-CDMA signals also leads to a low-rank matrix $\mathbf{S}$.

Generally, a limited amount of Kronecker products can be useful for the approximation of smooth and periodic signals or functions [51]. The low rank property can be exploited to allow various compact representations. This is essential in, e.g., large-scale blind signal separation with a large amount of sensors [51], or in multivariate function approximation where the number of function values exceeds the number of atoms in the universe [170, 214].

Tensors obtained from segmentation and decimation

The segmentation (decimation) procedure can be applied again on each column of $\mathbf{S}$ ($\mathbf{D}$). By stacking the obtained matrices along a third mode, a third-order tensor is obtained. This can be done a number of times to obtain a tensor $\mathcal{S}$ ($\mathcal{D}$) of order D. For example, time series consisting of molecule concentration values from multivariate curve resolution can be segmented to a third-order tensor, of which the modes indicate the specific hour, day and year [13, 14].

Again, $\mathcal{S}$ has rank 1 if and only if $\mathbf{f}$ can be written as the Kronecker product of D non-zero vectors, i.e., $\mathbf{f} = \mathbf{u}^{(D)} \otimes \cdots \otimes \mathbf{u}^{(1)}$.

Insert 12 — Quantization: In (very) large-scale applications, a compact representation with a low amount of parameters is often desired to reduce the computation time. Given a vector $\mathbf{f} \in \mathbb{K}^N$, let us apply $D-1$ segmentation

steps to obtain a tensor $\mathcal{S} \in \mathbb{K}^{q \times q \times \cdots \times q}$ of order D, with $N = q^D$. Using a small segment length q and large order D, this process is called *q-adic folding* or *quantization* in tensor-based scientific computing [170, 214, 215, 285]. $\mathcal{S}$ is then approximated using TT decompositions (leading to quantized TT or qTT), hierarchical Tucker decompositions or low-rank CPDs.

Let us focus on low-rank CPD, by way of example. The number of degrees of freedom in a rank-R CPD of $\mathcal{S}$ is equal[9] to $(qD - D + 1)R$. For fixed R, a maximal compression of $\mathbf{f}$ is obtained for $q = 2$, leading to only $(1 + \log_2 N)R$ degrees of freedom or representation parameters. This results in a significant compression compared to the original vector length N. Note that the case $q = 2$ is considered in Insert 10 as well, in the context of Hankelization.

Segmentation with overlap and/or with alignment

Hankelization maximally exploits shift invariance, while segmentation offers a maximally compact representation. A compromise between both can be made by selecting a larger subset of columns of the Hankel matrix than in basic segmentation. This amounts to stacking overlapping segments of a given vector.

Another variant consists of the extraction and stacking of non-consecutive segments. When tensorizing electrocardiogram (ECG) signals, for example, the segments may be chosen such that the heartbeats are aligned [54, 167, 168].

2.6.3 Löwner matrices and tensors

Löwner matrices [259] are intimately connected with rational functions, i.e., functions that can be written as a fraction of two polynomials. They have been used in system identification [22, 23]. Remarkably, unlike in the Hankel case, equidistant sampling is not required to obtain low-rank Löwner matrices.

Löwner matrix

Consider a vector $\mathbf{f} \in \mathbb{K}^N$, obtained from the evaluation of a function $\underline{f}(t)$ in the points $\{t_1, \ldots, t_N\}$. Let us construct a matrix $\mathbf{Z} \in \mathbb{K}^{N \times N}$ with $z_{ij} = \frac{f_i - f_j}{t_i - t_j}$.

[9] Note that there are qDR variables while the scaling indeterminacy leads to $(D-1)R$ degrees of freedom.

This definition would result in poorly defined fractions[10] on the diagonal of $\mathbf{Z}$, however. Löwner matrices are defined as proper submatrices of $\mathbf{Z}$.

More formally, let us partition the point set $T = \{t_1, \ldots, t_N\}$ in the point sets $X = \{x_1, \ldots, x_I\}$ and $Y = \{y_1, \ldots, y_J\}$, with $N = I + J$. Let vectors $\boldsymbol{\phi} \in \mathbb{N}^I$ and $\boldsymbol{\theta} \in \mathbb{N}^J$ contain the indices of the points from X and Y within T, respectively, e.g., $x_3 = t_{\phi_3}$. The entries of the Löwner matrix $\mathbf{L} \in \mathbb{K}^{I \times J}$ are then defined as $\frac{f_{\phi_i} - f_{\theta_j}}{x_i - y_j}$ for $1 \leq i \leq I$, $1 \leq j \leq J$, leading to the following matrix:

$$\mathbf{L} = \begin{bmatrix} \frac{f_{\phi_1} - f_{\theta_1}}{x_1 - y_1} & \frac{f_{\phi_1} - f_{\theta_2}}{x_1 - y_2} & \cdots & \frac{f_{\phi_1} - f_{\theta_J}}{x_1 - y_J} \\ \frac{f_{\phi_2} - f_{\theta_1}}{x_2 - y_1} & \frac{f_{\phi_2} - f_{\theta_2}}{x_2 - y_2} & \cdots & \frac{f_{\phi_2} - f_{\theta_J}}{x_2 - y_J} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{f_{\phi_I} - f_{\theta_1}}{x_I - y_1} & \frac{f_{\phi_I} - f_{\theta_2}}{x_I - y_2} & \cdots & \frac{f_{\phi_I} - f_{\theta_J}}{x_I - y_J} \end{bmatrix}. \tag{2.13}$$

Independent of the type of partitioning, it can be easily seen that $\mathbf{L}$ has rank 1 if[11] $\mathbf{f}$ is a sampled rational function of degree 1. Indeed, if $f(t) = c\frac{t-n}{t-p}$, then $l_{ij} = \alpha \cdot \frac{1}{x_i - p} \cdot \frac{1}{y_j - p}$ with constant $\alpha = -c(p-n)^2$ and we have

$$\mathbf{L} = \alpha\, \mathbf{a}\mathbf{b}^{\mathrm{T}} \quad \text{with} \quad \begin{cases} \mathbf{a} = \left[\frac{1}{x_1 - p}, \ldots, \frac{1}{x_I - p}\right]^{\mathrm{T}} \in \mathbb{K}^I, \\ \mathbf{b} = \left[\frac{1}{y_1 - p}, \ldots, \frac{1}{y_J - p}\right]^{\mathrm{T}} \in \mathbb{K}^J. \end{cases}$$

The factor vectors contain sampled rational functions of degree 1. Note the similarity with the Hankel case in (2.11).

Let us now consider a general rational function of degree[12] R. As a degree-R rational function with distinct poles can be written as a sum of R degree-1 rational functions (partial fractions), any corresponding Löwner matrix has rank R [21, 32, 259]. The factor matrices have Cauchy structure, and the decomposition is known as the Cauchy decomposition [124, 152, 381, 382].

Note that polynomials, as a special case of rational functions, also lead to low-rank Löwner matrices. More specifically, a polynomial of degree Q yields a Löwner matrix of rank Q.

[10] Asymptotically, the diagonal entries may be defined as the first-order derivatives of $f(t)$ in the points $\{t_1, \ldots, t_N\}$.

[11] We want to point out that this condition is only necessary and not sufficient. Indeed, only if every possible partitioning of T into sets X and Y yields a rank-1 Löwner matrix, $\mathbf{f}$ is an evaluated rational function of degree 1.

[12] The degree of an irreducible rational function is defined as the maximum of the degrees of the polynomial in the numerator and the polynomial in the denominator.

Löwner tensors

To the best of the authors' knowledge, higher-order generalizations of a Löwner matrix have not been defined or investigated in the literature. A third-order Löwner tensor can be obtained by *Löwnerizing* each row or column of a Löwner matrix and by stacking the resulting matrices. Applying a number of Löwnerization steps results in a $I_1 \times I_2 \times \cdots \times I_D$ tensor of order D such that $N = \sum_d I_d$.

Löwner tensors admit a definition in a closed form. Consider a partition of T in D sets $X^{(d)} = \left\{x_1^{(d)}, x_2^{(d)}, \ldots, x_{I_d}^{(d)}\right\}$, each associated with an index vector $\phi^{(d)} \in \mathbb{N}^{I_d}$ such that $x_{i_d} = t_{\phi_{i_d}^{(d)}}$. The entries are defined for all indices as follows:

$$l_{i_1 i_2 \ldots i_D} = \sum_{d=1}^{D} \left[\frac{f_{\phi_{i_d}^{(d)}}}{\prod\limits_{\substack{m=1 \\ m \neq d}}^{D} \left(x_{i_d}^{(d)} - x_{i_m}^{(m)} \right)} \right]. \tag{2.14}$$

Similar to Löwner matrices, higher-order Löwner tensors have low (multilinear) rank for rational functions and polynomials. Whereas a Löwner matrix contains first-order finite differences, a Löwner tensor of order D contains finite differences of order $D-1$. It can be seen that (2.14) simplifies to (2.13) for $D = 2$, with $l_{ij} = \frac{f_{\phi_i}}{x_i - y_j} + \frac{f_{\theta_j}}{y_j - x_i}$. In [121], it is demonstrated that a rational function of degree R with distinct poles leads to a Löwner tensor of rank R for large enough dimensions.

Link with Hankelization

There exists a close link between Löwner matrices and Hankel matrices, with a one-to-one relationship between Löwner matrices of size $I \times J$ and Hankel matrices of the same size. The isomorphism consists of the mapping $\mathbf{H} \mapsto \mathbf{L} = \mathbf{W}_{\mathbf{x}} \mathbf{H} \mathbf{W}_{\mathbf{y}}^{\mathrm{T}}$ with (typically ill-conditioned) matrices $\mathbf{W}_{\mathbf{x}} \in \mathbb{K}^{I \times I}$ and $\mathbf{W}_{\mathbf{y}} \in \mathbb{K}^{J \times J}$ that depend on the point sets X and Y, respectively, and not on the vector $\mathbf{f}$ [124, 152, 381, 382].

Efficient tensor decompositions

Similar to the Hankel case, the structure in Löwner matrices and tensors can be exploited in the case of equidistant points [389, 390]. In this way, the explicit construction of the Löwner matrix/tensor can be avoided.

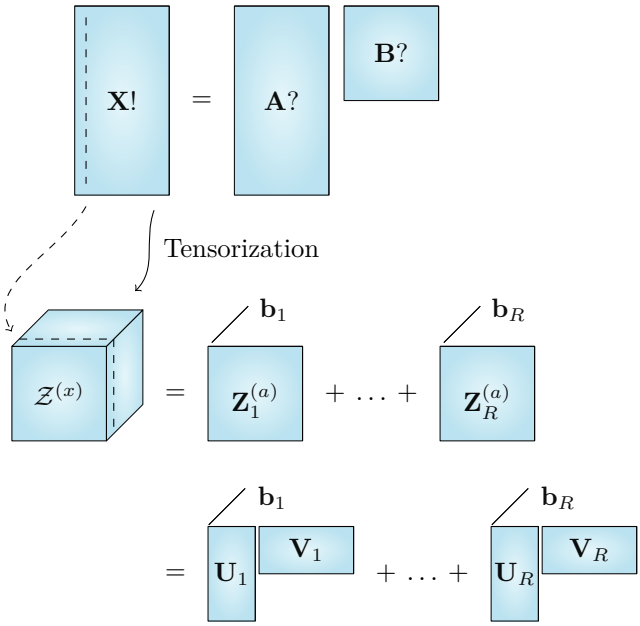


Figure 2.8: One deterministic approach for BSS is by mapping the matrix $\mathbf{X}$ to a tensor $\mathcal{Z}^{(x)}$ and by subsequently decomposing $\mathcal{Z}^{(x)}$ in multilinear rank-$(L_r, L_r, 1)$ terms. To preserve the linearity of the BSS model, a linear tensorization mapping should be used such that $\mathcal{Z}^{(x)}$ can be written as a linear combination of the contributions of the source signals. The vectors $\mathbf{b}_1, \ldots, \mathbf{b}_R$ appear as factor vectors in the third mode.

Insert 13 — Deterministic BSS: Hankelization, segmentation and Löwnerization can be used very effectively for deterministic blind signal separation (BSS) as discussed in [101], [51] and [124], respectively. It has been discussed in 1 that BSS and unique matrix factorization are connected.

Let us map each vector $\mathbf{x}_k$ (resp. $\mathbf{a}_r$) to a matrix $\mathbf{Z}_k^{(x)} \in \mathbb{K}^{I\times J}$ (resp. $\mathbf{Z}_r^{(a)} \in \mathbb{K}^{I\times J}$) using one of the methods. By stacking the matrices, a tensor $\mathcal{Z}^{(x)} \in \mathbb{C}^{I\times J\times K}$ (resp. $\mathcal{Z}^{(a)} \in \mathbb{C}^{I\times J\times R}$) is obtained as illustrated in Figure 2.8. Let us assume that each matrix $\mathbf{Z}_r^{(a)} = \mathbf{U}_r\mathbf{V}_r^{\mathrm{T}}$ has rank $L_r \leq \min(I, J)$, corresponding to the assumptions on $\mathbf{a}_r$ listed in Table 2.2. As each mapping is linear, we can write:

$$\mathcal{Z}^{(x)} = \mathcal{Z}^{(a)} \cdot_3 \mathbf{B} = \sum_{r=1}^{R} \mathbf{Z}_r^{(a)} \otimes \mathbf{b}_r = \sum_{r=1}^{R} (\mathbf{U}_r\mathbf{V}_r^{\mathrm{T}}) \otimes \mathbf{b}_r.$$

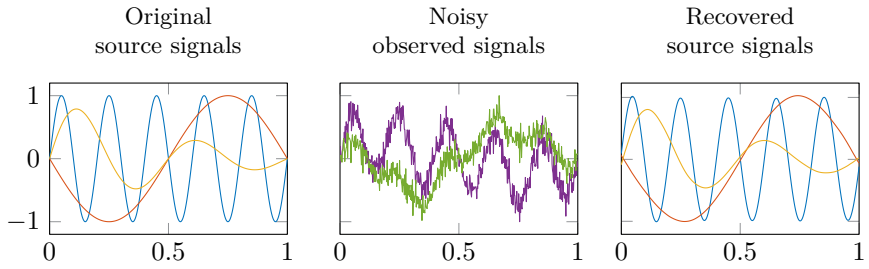


Figure 2.9: An example of Hankelization-based deterministic underdetermined BSS. Three signals (shown left, a high-frequency sinusoid (——), a low-frequency sinusoid (——) and an exponentially damped sinusoid (——)) are mixed together into two signals (middle, —— and ——). The signal-to-noise ratio of the observed signals is 10 dB, defined as the power of the signal to the power of Gaussian additive noise. Given only 500 samples per observed signal, the goal is to recover the source signals under the assumption that their corresponding Hankel matrices have low rank. On the right, the recovered source signals are shown, using $L_r = 2$ for each source signal. An optimal scaling and permutation step is added to cope with the default BSS indeterminacies. Note that the signals have been recovered nearly exactly given only 500 samples and despite this noisy underdetermined situation.

This boils down to a multilinear rank-$(L_r, L_r, 1)$ decomposition of $\mathcal{Z}^{(x)}$. Under mild conditions on $\mathbf{B}$, $\mathbf{U}_r$ and $\mathbf{V}_r$, this decomposition is unique and $\mathbf{B}$ can be directly recovered up to permutation and scaling [51, 101, 124]. An estimate of $\mathbf{A}$ can be obtained from $\mathbf{X}\big(\mathbf{B}^{\dagger}\big)^{\mathrm{T}}$, but also from $\mathcal{Z}^{(a)}$. Only the latter strategy succeeds in the underdetermined case. An illustration is given in Fig. 2.9. Note that the tensorization can also be applied on the second mode of $\mathbf{X}$ instead, if the mixing vectors rather than source signals satisfy one of the conditions from Table 2.2. The application of a tensorization technique together with a decomposition in multilinear rank-$(L_r, L_r, 1)$ terms has been referred to as block component analysis (BCA) [102].

The techniques have been applied in various applications. For example, Hankelization-based separation is well known from direction-of-arrival (DOA) estimation based on uniform linear arrays (ULA) [323], while segmentation is suitable for large-scale far- and near-field ULA-based DOA estimation [51]. Generalizations to blind system identification have appeared in [52]. Hankelization has also appeared in the context of separating epileptic seizures [199]. Löwnerization has been applied in the context of amino acid spectra separation. The tensor obtained by stacking excitation–emission spectra

Table 2.2: Overview of functions that may lead to low-rank matrices/tensors after tensorization

Tensorization	Functions
Hankelization	Exponentials, (exponentially damped) sinusoids, (exponential) polynomials, sum of exponentials
Segmentation	The functions above, periodic signals, sum of Kronecker products
Löwnerization	Rational functions, polynomials

of multiple mixtures may admit a CPD [57]. Given a single mixture, Löwnerization of the excitation–emission matrix can yield a unique CPD as well for spectra that can be well approximated by low-degree rational functions [124].

2.6.4 Determinant-defining matrix and monomial relations

A relatively new type of matricization exploits monomial relations between signal samples or data vector entries [339]. Various types of signals satisfy such relations, as we will show. We will link monomial relations with rank-deficient matrices and discuss tensor-based approaches for blind signal separation.

Monomial relations and rank-deficient matrices

Given a vector $\mathbf{f} \in \mathbb{C}^I$, let us assume there exist indices $p_1, \ldots, p_Q$ and $s_1, \ldots, s_T$, and integer powers $\beta_1, \ldots, \beta_Q$ and $\omega_1, \ldots, \omega_T$, such that

$$f_{p_1}^{\beta_1} f_{p_2}^{\beta_2} \cdots f_{p_Q}^{\beta_Q} = f_{s_1}^{\omega_1} f_{s_2}^{\omega_2} \cdots f_{s_T}^{\omega_T}. \tag{2.15}$$

Without loss of generality we assume that the degree of the right monomial is equal to or smaller than the degree of the left monomial, i.e., $L = \beta_1 + \ldots + \beta_Q =$

$\omega_0 + \omega_1 + \ldots + \omega_S$ for some $\omega_0 \geq 0$. Let us now construct two vectors:

$$\mathbf{b} = \begin{bmatrix} f_{p_1}\mathbf{1}_{\beta_1} \\ \vdots \\ f_{p_Q}\mathbf{1}_{\beta_Q} \end{bmatrix} \in \mathbb{K}^L \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} \mathbf{1}_{\omega_0} \\ f_{s_1}\mathbf{1}_{\omega_1} \\ \vdots \\ f_{s_T}\mathbf{1}_{\omega_Q} \end{bmatrix} \in \mathbb{K}^L.$$

It can be seen that (2.15) is equivalent with $\prod_{l=1}^{L} b_l = \prod_{l=1}^{L} c_l$, which then leads to the following:

$$\begin{aligned} 0 &= \prod_{l=1}^{L} b_l - \prod_{l=1}^{L} c_l \\ &= b_1 \cdot \begin{vmatrix} b_2 & 0 & \cdots & 0 \\ c_3 & \ddots & \ddots & \vdots \\ & \ddots & \ddots & 0 \\ & & c_L & b_L \end{vmatrix} - c_1 \begin{vmatrix} c_2 & b_2 & & \\ 0 & c_3 & \ddots & \\ \vdots & \ddots & \ddots & b_{L-1} \\ 0 & \cdots & 0 & c_L \end{vmatrix} \\ &= |\mathbf{D}|\,, \end{aligned} \tag{2.16}$$

with the $L \times L$ matrix $\mathbf{D}$ defined as

$$\mathbf{D} = \begin{bmatrix} b_1 & 0 & \cdots & 0 & (-1)^L \cdot c_1 \\ c_2 & b_2 & \ddots & & 0 \\ 0 & c_3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & c_L & b_L \end{bmatrix}. \tag{2.17}$$

The first transition is based on the fact that the determinant of a triangular matrix is equal to the product of the diagonal entries. Second, (2.16) can be seen as the cofactor expansion of $|\mathbf{D}|$ along the first row.

We have effectively matricized the data vector $\mathbf{f}$ to a rank-deficient matrix $\mathbf{D}$. Indeed, (2.15) leads to $|\mathbf{D}| = 0$. Hence, the rank of $\mathbf{D}$ is strictly smaller than L. On the other hand, the minors in (2.16) do not vanish except in the trivial case where both sides of (2.15) are zero. This means that the rank of $\mathbf{D}$ is exactly $L - 1$.

Examples

Consider an exponential function $f(t) = cz^t$ sampled in $I = 3$ equidistant points resulting in the data vector $\mathbf{f} \in \mathbb{K}^3$. It can be easily seen that $f_1 f_3 = f_2^2$. We

construct $\mathbf{b} = \begin{bmatrix} f_1, f_3 \end{bmatrix}^{\mathrm{T}}$, $\mathbf{c} = \begin{bmatrix} f_2, f_2 \end{bmatrix}^{\mathrm{T}}$ and the 2×2 matrix $\mathbf{D}$:

$$\mathbf{D} = \begin{bmatrix} b_1 & c_1 \\ c_2 & b_2 \end{bmatrix} = \begin{bmatrix} f_1 & f_2 \\ f_2 & f_3 \end{bmatrix},$$

which is a rank-deficient 2×2 Hankel matrix. A second example concerns 4-QAM signals, of which all samples satisfy the property $f^4 = -1$. Two 4-QAM samples $\mathbf{f} \in \mathbb{C}^2$ then satisfy $f_1^4 = f_2^4$ and we obtain the following rank-deficient matrix $\mathbf{D}$:

$$\mathbf{D} = \begin{bmatrix} f_1 & 0 & 0 & f_2 \\ f_2 & f_1 & 0 & 0 \\ 0 & f_2 & f_1 & 0 \\ 0 & 0 & f_2 & f_1 \end{bmatrix}.$$

BPSK, constant modulus, constant power and other finite alphabet signals satisfy similar monomial relations as well.

Data vectors satisfying multiple monomial relations

The entries of a vector $\mathbf{f} \in \mathbb{K}^N$ may satisfy more than one monomial relation. For example, let $\mathbf{f}$ be an equidistantly sampled exponential vector of length N. Among other relations, this vector satisfies $N-2$ relations of the type $f_{n-1} f_{n+1} = f_n^2$ for $n = 2, \ldots, N-1$. Imposing all monomial relations corresponds to constraining all 2×2 minors of $\mathbf{H}$ in (2.10) to be zero, which in its turn is equivalent with assuming that $\mathbf{H}$ has rank 1.

Second, N samples from a 4-QAM signal satisfy $Z = \frac{I(I-1)}{2}$ relations of the type $f_i^4 = f_j^4$ for $1 \leq i < j \leq N$, yielding Z different rank-deficient matrices of size $L \times L$.

A vector might exhibit multiple types of monomial relations which lead to matrices of different size. These matrices can be further processed using coupled matrix or tensor techniques [339]., matrices of possibly different size can be constructed which can be further processed with coupled matrix or tensor decomposition techniques. This flexibility is useful when dealing with non-trivial signal processing setups, such as L-shaped and triangular-shaped arrays.

Relations with other methods

We have pointed out that Hankelization may yield a single low-rank matrix while the technique from this section may result in a set of rank-deficient matrices. Hankelization, on the one hand, exploits all the structure in $\mathbf{f}$ in a

rigid manner. On the other hand, monomial relations enable a more flexible approach. They can, for example, be used when an entry is missing, or when a signal is not equidistantly sampled. A link between Löwnerization and the currently described technique can also be established, as, e.g., degree-1 rational functions also relate to specific monomial relations.

Insert 14 — Blind source separation using rank-deficient matrices: Monomial relations can be well exploited in a context of array processing, harmonic retrieval and blind signal separation. For example, ULAs are associated with exponential mixing vectors in far-field situations, while constant modulus signals such as 4-QAM or BPSK signals are often used in telecommunications. Considering the bilinear factorization in (2.1), let us assume that each of the columns of $\mathbf{A}$ satisfy the same set of Z monomial relations. Let $\mathbf{D}_{\mathbf{x}_k}^{(z)}$ and $\mathbf{D}_{\mathbf{a}_r}^{(z)}$ denote the matrices constructed as in (2.17) from $\mathbf{x}_k$ and $\mathbf{a}_r$, respectively, corresponding to the zth monomial relation of degree L_z, for $1 \leq z \leq Z$. Furthermore, let us concatenate these matrices along a third mode to obtain the tensors $\mathcal{D}_{\mathbf{X}}^{(z)}$ and $\mathcal{D}_{\mathbf{A}}^{(z)}$, respectively. Since the mapping in (2.17) is linear, the slices of $\mathcal{D}_{\mathbf{X}}^{(z)}$ are linear combinations of the slices of $\mathcal{D}_{\mathbf{A}}^{(z)}$ such that

$$\mathcal{D}_{\mathbf{X}}^{(z)} = \sum_{r=1}^{R} \mathbf{D}_{\mathbf{a}_r}^{(z)} \otimes \mathbf{b}_r,$$

for $1 \leq z \leq Z$. As each $L_z \times L_z$ matrix $\mathbf{D}_{\mathbf{a}_r}^{(z)}$ has rank $L_z - 1$, the tensors $\mathcal{D}_{\mathbf{X}}^{(z)}$ admit coupled decompositions in multilinear rank-$(L_z, L_z, 1)$ terms with a coupled factor matrix $\mathbf{B}$. As these decompositions are unique under mild conditions [339, 345], the mixing vectors and/or source signals can be recovered.

2.6.5 Time–frequency and time–scale techniques

In contrast to frequency domain representations, time–frequency analysis inspects the energy or intensity of a time series as a function of both time and frequency [87]. It can be used effectively to investigate non-stationary signals with time-evolving spectral characteristics, for example. We limit the discussion to the short-time Fourier transform and the Wigner–Ville distribution which are treated in Sections 2.6.5 and 2.6.5, respectively.

An alternative for time–frequency analysis is time–scale analysis. Instead of analyzing the signal using an exponential kernel as in the time–frequency case,

a single time-limited wave is used, also known as a wavelet. The wavelet decomposition, as will be discussed in 2.6.5, compares the signal to time-delayed and time-scaled versions of a given wavelet.

Both time–frequency and time–scale transforms map a vector to a matrix. If multiple signals are available, a tensor may be obtained by stacking the representations of the different signals. Space–time–frequency and space–time–scale analysis techniques consider a set of signals originating from a sensor or antenna array [256]. If the spatial mode rather than temporal mode is investigated, one refers to space–time–wave analysis [30, 31].

We want to mention that time–frequency and time–scale methods have a sparsifying effect for signals that have components that are well-located in both time and frequency or scale. This is the case for a variety of real-life time series and signals. For example, in epileptic encephalogram (EEG) data, epileptic seizures have significantly different spectral components and occur at different time instances than eye blinking and muscle tension artifacts [11, 199]. This sparsity is one of the reasons why the methods are frequently used in sparse coding, sparse component analysis (SCA) [60, 263] and blind signal separation [5, 305, 406, 415].

Short-time Fourier transform

One way to extract time-dependent spectral characteristics of a time signal $\underline{f}(t)$ is to consider the spectral content (Fourier transform) within small consecutive or overlapping intervals. This is called the Fourier transform with sliding window or short-time Fourier transform (STFT) [15]. Mathematically, the STFT is defined in the continuous domain as

$$\underline{w}(\omega, \tau) = \int_{-\infty}^{\infty} \underline{f}(t)\underline{h}(t-\tau)e^{-j\omega t}dt,$$

with angular frequency ω and window function $\underline{h}(t)$, commonly a Hann, Hamming or Gaussian[13] window. Note that the squared magnitude of the STFT, i.e., $|\underline{w}(\omega, \tau)|^2$, yields the well-known spectrogram [221].

In the discrete case, there is a clear link between the STFT and the segmentation technique discussed in Section 2.6.2 as different segments are extracted from the signal and then transformed using a discrete Fourier transform (DFT). This can be understood as follows. Let $\mathbf{f} \in \mathbb{K}^N$ and $\mathbf{h} \in \mathbb{K}^L$ be sampled versions of $\underline{f}(t)$ and $\underline{h}(t)$, respectively, using the same equidistant points. The window function $\underline{h}(t)$ is typically designed such that it is approximately zero outside the sampling

[13]The STFT with Gaussian window is also known as the Gabor transform.

interval. Let each column of $\mathbf{E} \in \mathbb{K}^{L \times J}$ consist of a length-L segment of $\mathbf{f}$. In the STFT, one typically considers partially overlapping segments rather than consecutive segments. The discrete STFT matrix $\mathbf{W} \in \mathbb{K}^{I \times J}$ with I uniformly spaced angular frequencies can then be written as

$$\mathbf{W} = \mathbf{Q}\,\mathrm{diag}(\mathbf{h})\mathbf{E},$$

with $\mathbf{Q} \in \mathbb{K}^{I \times L}$ the I-point DFT matrix.

Assuming that $I \geq L$, both $\mathbf{Q}$ and $\mathrm{diag}(\mathbf{h})$ have full column rank, and it follows that the rank of $\mathbf{W}$ is equal[14] to the rank of $\mathbf{E}$. As $\mathbf{E}$ is a segmented version of $\mathbf{f}$, the STFT completely inherits the low-rank properties from the segmentation tensorization technique of Section 2.6.2, independently of the choice of window function $\underline{h}(t)$. This theoretical connection with segmentation validates the use of low-rank approximations for STFT-based time–frequency representations.

It follows that in BSS methods relying on segmentation, such as in the separation of sums-of-exponentials discussed in Insert 13, the segmentation can be substituted by an STFT, and vice versa. For example, a combination of an STFT and CPD has been used for the separation of EEG data [269, 273] and audio data [288]. Also other tensor decompositions have been applied in combination with STFT, e.g., in spreading-multiplexing methods for blind system identification [16]. If segments overlap almost entirely, the STFT is strongly connected with Hankelization following the same rationale.

Quadratic representations — Autocorrelation function and Wigner–Ville distribution

The nonstationary symmetric autocorrelation function of a given zero-mean time signal $\underline{f}(t)$ is defined as $\underline{c}(t,\tau) = \underline{f}\left(t+\frac{\tau}{2}\right)\underline{f}\left(t-\frac{\tau}{2}\right)^{\star}$. The Wigner–Ville distribution (WVD) is obtained after applying a Fourier transform on the variable τ [395, 398]:

$$\underline{w}(\omega,t) = \int_{-\infty}^{\infty} \underline{f}\left(t+\frac{\tau}{2}\right)\underline{f}\left(t-\frac{\tau}{2}\right)^{\star} e^{-j\omega\tau} d\tau,$$

which is a function of the angular frequency ω and the lag τ.

The WVD $\underline{w}(\omega,t)$ is not a linear but a quadratic function of the input signal $\underline{f}(t)$. Hence, WVDs can theoretically not be used to solve the case studies described in this section such as pole estimation and blind signal separation from Inserts 11 and 13, as each underlying component does not only contribute

[14] Although the rank is equal, the non-zero singular values of $\mathbf{E}$ and $\mathbf{W}$ can differ.

a rank-1 term but also contributes additional cross-terms. To diminish the effect of the latter, one often relies on smoothing kernels, giving rise to the broad class of Cohen's quadratic time–frequency distributions [87]. While a detailed discussion of this class is outside the scope of this chapter, WVDs can be understood as the most simple forms. The Choi–Williams [80] and Kirkwood–Rihaczek distributions [220, 306] are other well-known quadratic representations, as well as the spectrogram discussed in the previous section.

By considering cross-distributions between signals, the cross-WVD and other quadratic representations can be used meaningfully in a tensor framework as they become linear in each signal. This is discussed in more detail in Section 2.7.1.

Wavelet decomposition

The STFT investigates low- and high-frequency components using the same fixed segment length. This can limit the spectral analysis if these components appear in broad and narrow parts of the signal, respectively, as is the case for various kinds of signals. It might be worthwhile, for example, to consider smaller segments for components with higher frequency so to obtain a better time resolution, i.e., such that high-frequency components can be better distinguished in the time domain. Time–scale analysis offers a solution by comparing the signal with dilated versions of a wavelet. The continuous wavelet decomposition (CWT) [96, 97, 265, 394] acts as such a representation, defined as

$$\underline{w}(\sigma, \tau) = \int_{-\infty}^{\infty} \underline{f}(t) \frac{1}{\sqrt{\sigma}} \underline{\psi} \left(\frac{t - \tau}{\sigma} \right)^{\star} dt, \tag{2.18}$$

in which the wavelet function $\underline{\psi}(t)$ (such as the Haar or Daubechies wavelet) plays a similar role as the STFT window, but is both translated and dilated. Figure 2.10 compares the time and frequency resolution of the STFT and CWT; note that Heisenberg's uncertainty principle prohibits both high time and frequency resolution [265].

Given a sampled time signal $\mathbf{f} \in \mathbb{K}^N$, a discretized CWT can readily be computed from (2.18). Again, a link with segmentation and Hankelization emerges as the discretized CWT $\mathbf{W} \in \mathbb{K}^{I \times J}$ of $\mathbf{f}$ can be written as

$$\mathbf{W} = \mathbf{Z}^{\mathrm{H}} \mathbf{E}, \tag{2.19}$$

with $\mathbf{Z} \in \mathbb{K}^{M \times I}$ and $\mathbf{E} \in \mathbb{K}^{M \times J}$. Each column of $\mathbf{E}$ is a length-M segment of $\mathbf{f}$, while each column of $\mathbf{Z}$ is a dilated version of the wavelet $\underline{\psi}(t)$, padded with zeros. The segment length M depends on the largest chosen scale. The formulation distinguishes between the signal component and translation effect

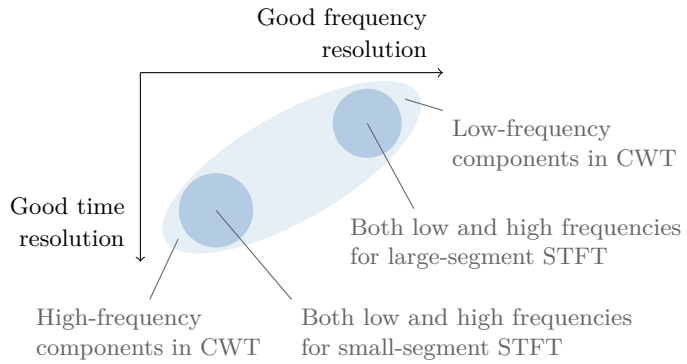


Figure 2.10: Using small segments, the STFT (▬) offers a good time resolution. It offers a better frequency resolution but inferior time resolution if large segments are used. The CWT (▬) offers a good time and frequency resolution for high- and low-frequency components, respectively.

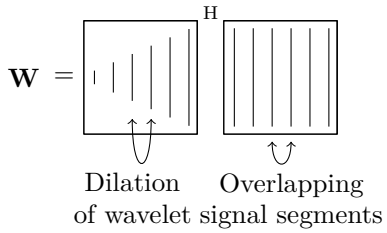


Figure 2.11: Illustration of the discrete wavelet decomposition in (2.19). The matrix $\mathbf{Z}$ contains dilated versions of the wavelet $\underline{\psi}(t)$ and the matrix $\mathbf{E}$ contains segments of the given signal.

on one hand (captured in $\mathbf{E}$), and the wavelet component and dilation effect on the other hand (captured in $\mathbf{Z}$). Figure 2.11 illustrates the structure of (2.19). More technically, each vector $\mathbf{z}_i$ is the finite difference of a decimated version corresponding to the ith scale (possibly padded with zeros) of the function $\underline{\phi}(k) = \int_{-\infty}^{k} \underline{\psi}(t)dt$.

Note that the wavelet transform is a linear function of the given signal. As $\mathbf{Z}$ has full column rank for $M \geq I$, the rank of $\mathbf{W}$ is equal to the rank of $\mathbf{E}$, and the discretized CWT inherits the low-rank properties from segmentation and Hankelization. For example, for a sinusoid, the matrix $\mathbf{W}$ has rank 2. Similar to the window function in the STFT, the type of wavelet alters the singular

values of $\mathbf{W}$.

Note that, during the construction of $\mathbf{E}$ in practice, the discrete signal $\mathbf{f}$ may be padded with zeros. This padding can create severe artifacts and may break the low-rank properties. To keep the low-rank properties, it suffices to remove a number of columns at the beginning and end of $\mathbf{W}$.

The discrete wavelet transform (DWT) provides an alternative for the discretized CWT. It uses $\underline{\psi}(\frac{t}{\sigma} - \tau)$ rather than $\underline{\psi}\left(\frac{t-\tau}{\sigma}\right)$. It can be seen that this change in variables breaks the low-rank properties of the discretized CWT. On the other hand, a signal that does not yield a low-rank discretized CWT matrix may yield a low-rank matrix after DWT. We omit a further discussion on the DWT. Note that the DWT formulation is equivalent to discrete filter banks, which will be briefly discussed in Section 2.6.6.

A choice can be made between different types of wavelets. While we have seen that the choice of wavelet does not alter the rank algebraically, it can influence the sparsifying effect. Different kinds of wavelets have been used in combination with different types of tensor decompositions. Morlet (also known as Gabor) wavelets, for example, have been used to construct a third-order tensor by stacking the CWTs of different EEG signals, after which a CPD has been applied [173, 269, 274, 397]. A Ricker or Mexican-hat CWT in combination with a CPD has been used as well [11, 238], while a decomposition in multilinear rank-$(L_r, L_r, 1)$ terms has been applied in, e.g., [199]. Biorthogonal CWTs have appeared in [117, 128] while tensor-based CWT processing has also been applied in hyperspectral imaging [251].

Note that a two-dimensional extension of the wavelet transform can also be applied on matrix data such as image data, possibly using separable two-dimensional wavelets [96]. A third-order tensor is obtained, of which two modes correspond to horizontal and vertical translations while a third mode corresponds to the wavelet dilation. Shearlets are anisotropic multiscale extensions of wavelets, naturally yielding fourth-order tensors [141].

2.6.6 Other tensorization techniques

The techniques presented in the Sections 2.6.1 to 2.6.5 are well-known linear mappings from vectors or matrices to tensors with established low-rank properties. There exist a number of other tensorization techniques as well, of which we briefly discuss three techniques. However, low-rank and other meaningful properties validating the use of these techniques in applications are yet to be identified.

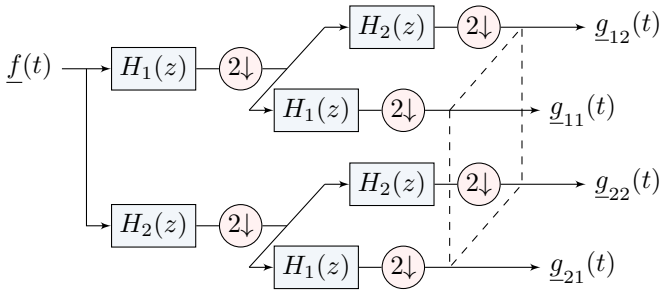


Figure 2.12: Uniform tree-structured filter bank in a tensor format.

A first example consists of Cauchy tensors. A vector $\mathbf{f}$ can be tensorized to an order-D Cauchy tensor $\mathcal{C}$ with entries $c_{i_1 \ldots i_D} = 1/(f_{i_1} + \cdots + f_{i_D})$ [302] or to other types of structured tensors.

Second, the empirical mode decomposition (EMD) extracts components recursively from a given vector $\mathbf{f} \in \mathbb{K}^N$ to obtain a matrix $\mathbf{E} \in \mathbb{K}^{N \times I}$ [196]. Let $\mathbf{e}_i$ be the ith so-called intrinsic mode function (IMF) of $\mathbf{f}$, then $\mathbf{e}_{i+1}$ is the mean of the upper and lower envelope of $\mathbf{e}_i$. An additional Hilbert transform yields the Hilbert–Huang decomposition [195]. Other recursive functions can be applied as well, leading to other mappings.

Uniform tree-structured filter banks form the basis of a tensorization technique as well [329, 371]. The output of the filter bank can be naturally stacked in a tensor, as illustrated in Figure 2.12. For example, the filters $H_1(z)$ and $H_2(z)$ can be low-pass and high-pass filters, respectively. More than two filters can be applied in each stage, the decimators can be omitted and more than two stages can be included. A connection can be made with the DWT from Section 2.6.5 [349].

2.7 Tensorization of a given set of vectors

The techniques presented in this section consider the availability of a set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K$. Provided that each vector has dimension N, these vectors can be stacked in the columns of a matrix $\mathbf{X}$ of size $N \times K$. Again, these vectors can be time series, evaluated functions or observations from stochastic variables, among others.

On one hand, we consider the construction of a set of matrices that depend on

a parameter. For a chosen set of parameter values, such as lags or frames, a tensor is obtained by stacking the constructed matrices corresponding to each parameter value. Sometimes, as discussed in Section 2.7.1, the obtained tensor readily admits to a CPD. In other cases, as discussed in Section 2.7.2, a CPD emerges only after some manipulation. It will become apparent that diagonality is only one of the properties leading to a CPD.

On the other hand, rather than artificially stacking matrices, a tensor with natural multiway character can be directly constructed. Such tensorization techniques can then lead to diagonal tensors such as the context of unique matrix factorization. Higher-order statistics, adjacency tensors and score functions are addressed in Section 2.7.3.

More mathematically, the techniques in Sections 2.7.1 and 2.7.2 are based on the evaluation of a scalar parameter-dependent multivariate function $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$ in the Cartesian product set $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$. A set of parameter values then yields a set of matrices which can readily be stacked in a third-order tensor. Each function $\underline{z}$ can be used in that respect, but is not necessarily meaningful. Similar to Section 2.6, the links between the vector properties and the properties of the obtained tensor will be discussed. In Section 2.7.3 one then considers a function $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ with $D \geq 3$.

2.7.1 Collection of matrices obtained by parameter variation

From a given matrix $\mathbf{X}$, a set of $K \times K$ matrices $\mathbf{Z}_1, \ldots, \mathbf{Z}_L$ is generated via a mapping that depends on a parameter. By stacking these matrices, a third-order tensor $\mathcal{Z} \in \mathbb{K}^{K \times K \times L}$ is obtained. For a chosen set of parameter values $\rho_1, \ldots, \rho_L$, the lth matrix $\mathbf{Z}_l$ is constructed by evaluating a function $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$ in $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ as discussed in Section 2.4.1, with parameter value ρ_l. Hence, the entry $z_{k_1 k_2 l}$ of $\mathcal{Z}$ is defined as $\underline{z}(\mathbf{x}_{k_1}, \mathbf{x}_{k_2}, \rho_l)$.

We discuss stacked outer products, stacked covariance matrices and stacked Hessian or Jacobian matrices in more detail. There exist other variants, such as the stacking of time-dependent adjacency matrices. The latter will be covered in Section 2.7.3.

Insert 15 — Unique matrix factorization: The techniques from this subsection have been successfully used for what we have called 'unique matrix factorization' in Insert 1. The approach then typically consists of the following. Let $\mathbf{Z}_l^{(x)}$ and $\mathbf{Z}_l^{(a)}$ be constructed from $\mathbf{X}$ and $\mathbf{A}$, respectively. We limit ourselves to mappings $\underline{z}$ that are bilinear, i.e., $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$ is linear

in $\mathbf{u}^{(1)}$ and in $\mathbf{u}^{(2)}$. Thanks to the bilinearity, we have

$$\begin{aligned} z_{k_1k_2l} &= \underline{z}(\mathbf{x}_{k_1}, \mathbf{x}_{k_2}, \rho_l) \\ &= \underline{z}(\sum_{r_1} b_{k_1r_1}\mathbf{a}_{r_1}, \sum_{r_2} b_{k_2r_2}\mathbf{a}_{r_2}, \rho_l) \\ &= \sum_{r_1,r_2} b_{k_1r_1}b_{k_2r_2}\underline{z}(\mathbf{a}_{r_1}, \mathbf{a}_{r_2}, \rho_l) \end{aligned}$$

such that we can write $\mathcal{Z}^{(x)} = \mathcal{Z}^{(a)} \bullet_1 \mathbf{B} \bullet_2 \mathbf{B}$ or

$$\begin{aligned} \mathbf{Z}_1^{(x)} &= \mathbf{B}\mathbf{Z}_1^{(a)}\mathbf{B}^{\mathrm{T}}, \\ &\vdots \\ \mathbf{Z}_L^{(x)} &= \mathbf{B}\mathbf{Z}_L^{(a)}\mathbf{B}^{\mathrm{T}}. \end{aligned}$$

If the matrices $\mathbf{Z}_l^{(a)}$ can be assumed diagonal, the tensor $\mathcal{Z}^{(x)}$ admits a rank-R CPD $\mathcal{Z}^{(x)} = [\![\mathbf{B}, \mathbf{B}, \mathbf{D}^{(a)}]\!]$, as illustrated in Figure 2.13. The matrix $\mathbf{D}^{(a)}$ consists of the diagonals of the matrices $\mathbf{Z}_l^{(a)}$. The uniqueness of the CPD of $\mathcal{Z}^{(x)}$ under mild conditions on $\mathbf{B}$ and $\mathbf{D}^{(a)}$ is key to the unique factorization of the matrix $\mathbf{X}$. Note that the mapping $\underline{z}$ is chosen in function of prior knowledge on $\mathbf{A}$, as it translates known properties of $\mathbf{A}$ into diagonality of $\mathbf{Z}_l^{(a)}$.

Stacked outer products

Let us take $\underline{z}$ equal to the following function:

$$\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho) = u_\rho^{(1)} u_\rho^{(2)},$$

with $\rho \in \{1, \ldots, N\}$. Hence, if we vary ρ from 1 to N, it can be seen that the nth matrix $\mathbf{Z}_n \in \mathbb{K}^{K \times K}$ is the symmetric outer product of the nth row of $\mathbf{X}$ with itself. Here, the parameter ρ equals the row index n. By stacking all outer products, the tensor $\mathcal{Z} \in \mathbb{K}^{K \times K \times N}$ is obtained for which $\mathbf{Z}_{(3)} = \mathbf{X} \odot^{\mathrm{T}} \mathbf{X}$.

By definition, each slice of $\mathcal{Z}$ has rank 1. The overall tensor $\mathcal{Z}$ has rank 1 if and only if $\mathbf{x}_1, \ldots, \mathbf{x}_K$ are scaled versions of each other or, stated otherwise, if $\mathbf{X}$ has rank 1. However, the rank of $\mathcal{Z}$ does not equal the rank of $\mathbf{X}$ in general.

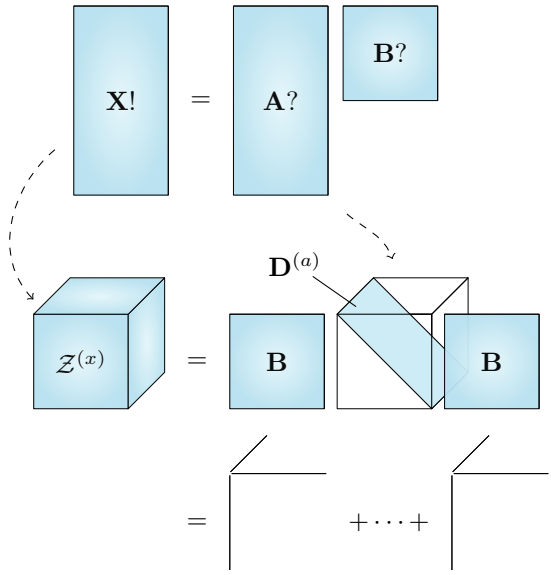


Figure 2.13: One approach to solve blind signal separation, or unique matrix factorization in general, consists of constructing a set of matrices depending on a parameter, which yields a tensor $\mathcal{Z}^{(x)}$. Provided a bilinear mapping is used, this tensor can be written as $\mathcal{Z}^{(x)} = \mathcal{Z}^{(a)} \bullet_1 \mathbf{B} \bullet_2 \mathbf{B}$. Joint diagonalization of the slices $\mathbf{Z}_1^{(x)}, \ldots, \mathbf{Z}_L^{(x)}$ is equivalent with the CPD of $\mathcal{Z}^{(x)}$.

This holds only if each row vector of $\mathbf{X}$ is a scaled version of one of the columns of a matrix $\mathbf{B} \in \mathbb{K}^{K \times R}$. Let us now consider the case in which every row of $\mathbf{X}$ contains exactly one nonzero entry. Then, and only then, we obtain diagonal[15] matrices $\mathbf{Z}_l$. One can think of such $\mathbf{X}$ as a (scaled) selection matrix, as the multiplication $\mathbf{X}\mathbf{Y}^{\mathrm{T}}$ 'selects' (and possibly scales) columns from an arbitrary matrix $\mathbf{Y}$.

It is obvious that third-order tensors $\mathcal{Z}_n$ constructed from a generalization of $\underline{z}$ can also be stacked, as well as higher-order generalizations. These generalizations still satisfy the rank and diagonality properties.

Insert 16 — Clustering: By assuming a selection matrix $\mathbf{A}$ in unique matrix factorization, we obtain a clustering model. Indeed, the columns of $\mathbf{B}$ then contain the cluster vectors, and the positions of the nonzero entries

[15] Furthermore, each diagonal consists of only a single nonzero entry. More specifically, the diagonal is equal to the corresponding element-wise squared row of $\mathbf{X}$. We ignore this in the remainder.

in the rows of $\mathbf{A}$ determine the cluster allocations, e.g.:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \mathbf{b}_4 \end{bmatrix}^{\mathrm{T}}.$$

As the structure in $\mathbf{A}$ allows us to assume that each matrix $\mathbf{Z}_n^{(a)}$ is diagonal, $\mathcal{Z}^{(x)}$ admits a rank-R CPD with

$$\mathcal{Z}^{(x)} = \left[\!\!\left[\mathbf{B}, \mathbf{B}, \mathbf{D}^{(a)} \right]\!\!\right]. \tag{2.20}$$

Furthermore, the diagonal of $\mathbf{Z}_n^{(a)}$ consists of squared entries of the nth row of $\mathbf{A}$. Hence, we have $\mathbf{D}^{(a)} = \mathbf{A} * \mathbf{A}$.[16] The CPD is unique under mild conditions on $\mathbf{A}$ and $\mathbf{B}$, and each cluster contributes a rank-1 term to $\mathcal{T}^{(x)}$. The condition on $\mathbf{A}$, for example, only requires that each column of $\mathbf{A}$ contains at least one nonzero entry, i.e., that from each cluster at least one observation is available.

In topic modeling, for example, each vector $\mathbf{b}_r$ collects the occurrence probability of a set of N dictionary words for the rth topic, and each vector $\mathbf{x}_k$ contains the occurrence frequency of the dictionary words in the kth document [18]. The goal then consists of allocating a topic to each document. It is equal to the 'bag-of-words' problem, and is a specific instance of latent Dirichlet allocation [47]. The latter allows mixed models in which documents can rely on more than one topic.

Instead of occurrence frequency or probability data, the given vectors can also be feature vectors. For example, in MRI processing, stacked outer products have been used to differentiate tumor tissue types, with each tissue type corresponding to a different feature vector and hence different cluster [46].

Note that the CPD formulation avoids the use of heuristic algorithms such as k-means [185]. In the presence of noise, algebraic CPD algorithms may still give suboptimal estimates of the matrix $\mathbf{A}$ and the cluster vectors in $\mathbf{B}$. If Gaussian noise is added to each row of $\mathbf{X}$ with statistics that depend on the specific cluster the row belongs to, the problem is equivalent to Gaussian mixture parameter estimation [157, 193]. Higher-order moments as discussed in Section 2.7.3 may be more appropriate because of their noise reduction abilities.

Stacked covariance matrices

Let now $\mathbf{x}_1, \ldots, \mathbf{x}_K$ be observations of K stochastic variables $\chi_1, \ldots, \chi_K$ with the latter being collected in a random vector $\boldsymbol{\chi} \in \mathbb{K}^K$. The covariance matrix of $\boldsymbol{\chi}$ is defined as $\mathbf{C}^{(x)} = \mathrm{E}\{\boldsymbol{\chi}(t)\boldsymbol{\chi}^{\mathrm{H}}(t)\}$. Given the observations $\mathbf{x}_1, \ldots, \mathbf{x}_K$, the sample covariance matrix $\hat{\mathbf{C}}^{(x)} = \frac{1}{N-1}\mathbf{X}^{\mathrm{H}}\mathbf{X}$ is an unbiased estimator of $\mathbf{C}^{(x)}$. The covariance matrix is diagonal if and only if the variables χ_k are uncorrelated. Consequently, it is also diagonal if the variables χ_k are statistically independent. However, the diagonality of $\mathbf{C}^{(x)}$ is not sufficient for independence. Hence, $\mathbf{C}^{(x)}$ by itself does not provide sufficient information for, e.g., independent component analysis (ICA) [89, 210], such that additional matrices need to be computed. These matrices can then be stacked such that tensor tools can be employed.

One may for instance compute a number of different covariance matrices. Several variants have been reported in the literature. Lagged covariance matrices, for example, compare time-delayed signal versions:

$$\mathbf{C}^{(x)}(\rho) = \mathrm{E}\{\boldsymbol{\chi}(t)\boldsymbol{\chi}^{\mathrm{H}}(t-\rho)\}.$$

Covariance matrices can also be estimated in function of the specified time frame. In the following, s_ρ and e_ρ denote the start and end of each time frame indicated by ρ, respectively:

$$\mathbf{C}^{(x)}(\rho) = \mathrm{E}\{\boldsymbol{\chi}(t)\boldsymbol{\chi}^{\mathrm{H}}(t)\}_{s_\rho \leq t < e_\rho}.$$

Instead of varying only the lag or frame index, both may be varied by considering time-frequency representations and selecting specific time–frequency working points, cf. Section 2.6.5. Indeed, one can for example consider the cross-Wigner–Ville distribution which is closely related to the covariance matrix. The selection of time–frequency points corresponds to the sparsifying effect of the representations [36]. For example, it can be useful to select those time–frequency points where at most one of the underlying signal components is active.

Furthermore, covariance matrices with different complex conjugate patterns can be stacked as well, e.g., $\mathrm{E}\{\boldsymbol{\chi}(t)\boldsymbol{\chi}^{\mathrm{T}}(t)\}$ and $\mathrm{E}\{\boldsymbol{\chi}(t)\boldsymbol{\chi}^{\mathrm{H}}(t)\}$.

Instead of covariance matrices, so-called eigenmatrices of fourth-order cumulants have also been considered. These matrices are defined as reshaped eigenvectors of an unfolded fourth-order cumulant. We will discuss higher-order cumulants in more detail in 2.7.3.

[16] A more technical way to recognize the CPD is as follows: $\mathbf{Z}_{(3)} = \mathbf{X}\odot^{\mathrm{T}}\mathbf{X} = (\mathbf{A}\mathbf{B}^{\mathrm{T}})\odot^{\mathrm{T}}(\mathbf{A}\mathbf{B}^{\mathrm{T}}) = (\mathbf{A}\odot^{\mathrm{T}}\mathbf{A})(\mathbf{B}^{\mathrm{T}}\otimes\mathbf{B}^{\mathrm{T}})$. Because of the structure in $\mathbf{A}$, $\mathbf{a}_i * \mathbf{a}_j = \mathbf{0}_N$ for $i \neq j$. By omitting the cross-terms, we obtain $(\mathbf{A}*\mathbf{A})(\mathbf{B}^{\mathrm{T}}\odot^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) = (\mathbf{A}*\mathbf{A})(\mathbf{B}\odot\mathbf{B})^{\mathrm{T}}$. The latter is an unfolded version of $[\![\mathbf{B}, \mathbf{B}, \mathbf{D}^{(a)}]\!]$ along the third mode with $\mathbf{D}^{(a)} = \mathbf{A} * \mathbf{A}$.

Insert 17 — Independent component analysis: The mentioned techniques have been used for ICA, in which one tries to recover a set of mutually independent source signals. Instantaneous ICA is a specific instance of unique matrix factorization from Insert 1, with $\mathbf{X}$ and $\mathbf{A}$ consisting of a set of observed signals and a set of source signals, respectively. $\mathbf{B}$ acts as mixing matrix. Each technique yields diagonal matrices for sets of independent signals such that for each technique a CPD is obtained as discussed in Insert 15.

However, each technique typically requires an additional working assumption. For example, lagged covariance matrices are only diagonal for temporally coherent independent signals. Such matrices are used in the algorithm for multiple unknown signals extraction (AMUSE) [360] which relies on a single lag[17] and in the second-order blind identification (SOBI) algorithm [33, 34] which uses multiple lags. For time-depending covariance matrices to be diagonal, on the other hand, it is required that the signals are non-stationary. These matrices have not only appeared in a signal processing context [295, 347] but also in a context of directed network topology inference [320]. By stacking covariance matrices with mutually different complex conjugate conventions, non-circular signals can be separated [108]. Furthermore, working with matrices which are evaluated in different time–frequency points typically involves the assumption of sparsity [35, 305, 406]. Finally, when using eigenmatrices of fourth-order cumulants, it is assumed that the source signals are non-Gaussian.

Note that besides each technique-dependent working assumption there are also typically mild algebraic conditions present, related to the uniqueness conditions of the CPD of $\mathcal{Z}^{(x)}$. In SOBI, for example, the source signals cannot be recovered if their autocorrelations are scaled versions of each other [33].

Stacked Hessian or Jacobian matrices

A tensor can be obtained by stacking Hessian or Jacobian matrices evaluated in different points, as discussed in Section 2.4.3. An example is the covariance matrix which is the Hessian matrix of the second characteristic function evaluated in the origin. Hence, all stacked matrices from Section 2.7.1 may be interpreted as stacked Hessian matrices. Note that the origin is merely a

[17] There is a close link between AMUSE and canonical correlation analysis as well [155, 192].

practical choice; other evaluation points can also be used but lead to more complex expressions [93, 405].

Instead of Hessian matrices, one can also stack Jacobian matrices. This has for instance been used for multivariate function decoupling, as discussed in Insert 3.

One might wonder how the concept of derivatives is related to the evaluation of a function $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$ as discussed in the beginning of this section. The answer is not straightforward, but will be given in Section 2.7.

2.7.2 Collection of matrices indirectly resulting in a CPD

In the previous section, a number of techniques have been reviewed which stack parameter-depending matrices. In various cases, these techniques have enabled the use of a CPD through the fact that the constructed matrices are diagonal for a specific set of vectors. For example, stacked outer products yield diagonal matrices for a given selection matrix, while covariance matrices are diagonal for independent signals. This diagonal property readily yields a CPD.

However, algorithms have appeared in which a CPD is obtained only after some manipulations. In this section, we will show that these algorithms can each be linked to the construction of a tensor by stacking different matrices. Instead of a diagonal property for a specific set of vectors, these matrices satisfy other kinds of properties.

Unlike in other sections, we will focus specifically on the problem of unique matrix factorization, i.e., the recovery of $\mathbf{A}$ and $\mathbf{B}$ given $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$. In the previous CPD appearances in applications such as deterministic BSS from Insert 13, clustering from Insert 16 and ICA from Insert 17, one or more of the factor matrices was equal to $\mathbf{B}$. Rather than $\mathbf{B}$, it can be seen that a so-called 'separation matrix' $\mathbf{W} = \left(\mathbf{B}^{-1}\right)^{\mathrm{T}}$ is recovered from the CPDs in this section such that $\mathbf{X}\mathbf{W} = \mathbf{A}$.

We start the discussion with an introductory case study on the separation of constant modulus signals in Section 2.7.2. We then generalize the procedure in Section 2.7.2 and provide an overview of the properties that also lead to a CPD after some manipulations, apart from the diagonal property. We then provide a number of examples of the technique in Section 2.7.2.

Case study – Separation of constant modulus signals

Let us assume that the entries of $\mathbf{A}$ have constant modulus, i.e., $|a_{nr}|^2 = a_{nr}a_{nr}^\star = c$, with $c \in \mathbb{R}$ the square of the constant modulus. Each observed vector $\mathbf{x}_k$ is an instantaneous mixture of the constant modulus signals $\mathbf{a}_1, \ldots, \mathbf{a}_R$ [384]. We assume $K = R$ but we continue to use both symbols to improve readability. If $K > R$ we can apply a dimension reduction technique such as PCA [208] to obtain the case $K = R$.

Similar to Section 2.7.1, we will apply a function $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$, namely

$$\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho) = u_\rho^{(1)} \left(u_\rho^{(2)}\right)^\star.$$

The parameter ρ corresponds to the sample index n. After evaluation in $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ and $\{\mathbf{a}_1, \ldots, \mathbf{a}_R\} \times \{\mathbf{a}_1, \ldots, \mathbf{a}_R\}$ for $n = 1, \ldots, N$, we obtain the tensors $\mathcal{Z}^{(x)} \in \mathbb{K}^{K \times K \times N}$ and $\mathcal{Z}^{(a)} \in \mathbb{K}^{R \times R \times N}$ with slices $\mathbf{Z}_n^{(x)}$ and $\mathbf{Z}_n^{(a)}$, respectively. The function $\underline{z}$ is linear in $\mathbf{u}^{(1)}$ and in $\mathbf{u}^{(2)}$ such that, as discussed in Insert 15, we can write

$$\mathcal{Z}^{(x)} = \mathcal{Z}^{(a)} \bullet_1 \mathbf{B} \bullet_2 \mathbf{B}^\star \tag{2.21}$$

or

$$\mathbf{Z}_1^{(x)} = \mathbf{B}\mathbf{Z}_1^{(a)}\mathbf{B}^{\mathrm{T}},$$

$$\vdots$$

$$\mathbf{Z}_N^{(x)} = \mathbf{B}\mathbf{Z}_N^{(a)}\mathbf{B}^{\mathrm{T}}.$$

In this case, unlike in Section 2.7.1 and Figure 2.13, the matrices $\mathbf{Z}_n^{(a)}$ are not diagonal. However, we know that each diagonal entry of $\mathbf{F}_n^{(a)}$ is constant and equal to c. Under this constraint, the problem in (2.21) can be solved in a classical optimization setting. On the other hand, it is clear that there is only little difference with solving the standard problem $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$ under the constant modulus constraint on $\mathbf{A}$. However, as we will now show, (2.21) does provide a basis to derive algebraic algorithms and uniqueness conditions.

Let us consider the previously defined separation matrix $\mathbf{W} \in \mathbb{K}^{K \times R}$ such that $\mathbf{A} = \mathbf{X}\mathbf{W}$. For each separation vector $\mathbf{w}_r$, we obtain a constant modulus vector $\mathbf{X}\mathbf{w}_r = \mathbf{a}_r$. From (2.21), it follows that

$$\mathcal{Z}^{(x)} \bullet_1 \mathbf{W}^{\mathrm{T}} \bullet_2 \mathbf{W}^{\mathrm{H}} = \mathcal{Z}^{(a)}. \tag{2.22}$$

Recalling that $\mathbf{Z}_{(3)}$ is the mode-3 unfolding of $\mathcal{Z}$, (2.22) is equivalent to

$$\mathbf{Z}_{(3)}^{(x)} (\mathbf{W}^\star \otimes \mathbf{W}) = \mathbf{Z}_{(3)}^{(a)}.$$

Let us only consider the equations that correspond to the diagonals of $\mathbf{Z}_n^{(x)}$ and $\mathbf{Z}_n^{(a)}$, and let $\mathbf{D}^{(a)}$ contain the diagonals of $\mathbf{Z}_n^{(a)}$ as in Section 2.7.1, i.e., $\mathbf{D}^{(a)} = \left[\operatorname{diag}(\mathbf{Z}_1^{(a)}), \ldots, \operatorname{diag}(\mathbf{Z}_N^{(a)})\right]^{\mathrm{T}} \in \mathbb{C}^{N \times R}$. Hence, we have

$$\mathbf{Z}_{(3)}^{(x)} \mathbf{Y} = \mathbf{D}^{(a)} = c\mathbf{1}_{N \times R}, \quad \text{with } \mathbf{Y} = \mathbf{W}^\star \odot \mathbf{W}. \tag{2.23}$$

This is a linear system in which each solution has a Kronecker structure, i.e., $\mathbf{w}_r^\star \odot \mathbf{w}_r$. This can be solved with suitable algebraic or optimization-based algorithms.

Let us go one step further and relax the structure of the columns of $\mathbf{Y}$ to reduce (2.23) to a homogeneous system. This will allow us to uncover a CPD in the null space of $\mathbf{Z}_{(3)}^{(x)}$ from which $\mathbf{W}$ can be estimated. By applying a Householder or discrete Fourier transform, among others, and omitting the first row, the problem can be easily transformed to

$$\tilde{\mathbf{Z}}_{(3)}^{(x)} \mathbf{Y} = \mathbf{0}_{N \times R} \tag{2.24}$$

with $\tilde{\mathbf{Z}}_{(3)}^{(x)} \in \mathbb{C}^{(N-1) \times K^2}$. From (2.24) we know that the dimension of the null space of $\tilde{\mathbf{Z}}_{(3)}^{(x)}$ is at least R. This corresponds to the fact that there are R separation vectors yielding R constant modulus signals. Let us assume that there are sufficient samples available such that the number of rows of $\tilde{\mathbf{Z}}_{(3)}^{(x)}$ is larger than $K^2 - R$, and that the dimension of the null space is exactly equal[18] to R. Let the columns of $\mathbf{N} \in \mathbb{C}^{K^2 \times R}$ form a basis of this null space. There exists a matrix $\mathbf{\Lambda} \in \mathbb{C}^{R \times R}$ such that $\mathbf{N} = \mathbf{Y}\mathbf{\Lambda}^{\mathrm{T}} = (\mathbf{W}^\star \odot \mathbf{W})\,\mathbf{\Lambda}^{\mathrm{T}}$. Hence, by reshaping $\mathbf{N}$ to the tensor $\mathcal{N} \in \mathbb{C}^{K \times K \times R}$, the rank-$R$ CPD $\mathcal{N} = [\![\mathbf{W}, \mathbf{W}^\star, \mathbf{\Lambda}]\!]$ is obtained. After estimating $\mathbf{W}$, both $\mathbf{A}$ and $\mathbf{B}$ can be recovered. This sequence of steps forms the basis of the well-known analytical constant modulus algorithm (ACMA) [384]. Note that this third method is fundamentally different from the second method, as solving (2.23) on the one hand and computing the CPD of $\mathcal{N}$ after the computation of the null space from (2.24) on the other hand both involve different weightings.

[18] This corresponds to $r(\tilde{\mathbf{Z}}_{(3)}^{(x)}) = K^2 - R$, which is a valid assumption if the phases of the source samples are sufficiently random, cf. [363, 384].

Generalization

The case study in the previous section leads us to an important novel concept: as long as a function $\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \rho)$ can be found that yields a tensor $\mathcal{F}^{(a)}$ of which the slice diagonals have constant entries, a CPD emerges provided a number of algebraic conditions are satisfied. This concept can be further generalized as we will now discuss. A CPD can also be obtained if the diagonals of $\mathbf{F}_l^{(a)}$ (cf. Figure 2.13) yield a zero matrix, a rank-1 matrix, or more generally, a matrix of rank at most $R-2$.

Let us first discuss the rank-1 variant. A Householder or discrete Fourier transform is not necessary to uncover a CPD from (2.23). Obviously, the constant matrix $c\mathbf{1}_{N\times R} = c\mathbf{1}_N\mathbf{1}_R^{\mathrm{T}}$ has rank 1; let us assume that $\mathbf{D}^{(a)} = \mathbf{g}\mathbf{h}^{\mathrm{T}}$ with $\mathbf{g} \in \mathbb{K}^N$ and $\mathbf{h} \in \mathbb{K}^R$. Hence, under the same conditions as before on the number of samples and the column rank of $\mathbf{Z}_{(3)}^{(x)}$, it can be assumed from (2.23) that the null space of $\mathbf{Z}_{(3)}^{(x)}$ has exactly dimension $R-1$. Let the columns of $\mathbf{N} \in \mathbb{K}^{K^2\times(R-1)}$ form a basis of this null space. By folding $\mathbf{N}$ to a tensor $\mathcal{N} \in \mathbb{K}^{K\times K\times(R-1)}$ as before, a rank-R CPD $\mathcal{N} = [\![\mathbf{W}, \mathbf{W}^\star, \mathbf{\Lambda}]\!]$ emerges again. Note that we do not need to know the values in $\mathbf{g}$ or $\mathbf{h}$ to obtain the CPD.

The rank-1 condition on $\mathbf{D}^{(a)}$ can be relaxed even further, as the rank of $\mathbf{D}^{(a)}$ can be up to $R-2$. In the latter case, as long as the algebraic conditions are satisfied, a tensor $\mathcal{N} \in \mathbb{K}^{R\times R\times 2}$ is obtained that admits a rank-R CPD. Although $\mathcal{N}$ consists of only two slices, the CPD can still be unique following from Kruskal's uniqueness conditions [230]. Note that the rank of $\mathbf{D}^{(a)}$ cannot be larger than $R-2$. If $r(\mathbf{D}^{(a)}) = R-1$, for example, the the null space has dimension one and folding a basis of this null space only yields a matrix rather than a third-order tensor.

On the other hand, instead of having rank 1 or higher rank, $\mathbf{D}^{(a)}$ can also immediately be a zero matrix, i.e., $\mathbf{D}^{(a)} = \mathbf{0}_{N\times R}$. A reduction to a homogeneous system as in (2.24) is then obviously not necessary. Folding a basis of the null space of $\mathbf{Z}_{(3)}^{(x)}$ readily yields a $K \times K \times R$ tensor $\mathcal{N}$ of rank R, provided the same conditions as before are satisfied. Note that $\mathbf{D}^{(a)} = \mathbf{0}_{N\times R}$ results in hollow matrices $\mathbf{Z}_n^{(a)}$, i.e., the diagonals of $\mathbf{Z}_n^{(a)}$ are zero. It is interesting to connect this to the techniques from Section 2.7.1 such as SOBI. There, the diagonal entries of $\mathbf{Z}_l^{(a)}$ were assumed to be non-zero while the off-diagonal entries were assumed to be zero. Here, it is exactly the opposite. As we have uncovered, both properties effectively lead to a CPD, whether directly or indirectly.

There is an important connection with the technique from Section 2.6.4. While the latter considers monomial relations, the function $\underline{z}$ in this section is related

to the broader class of polynomial relations. The vectors $\mathbf{a}_r$ are then assumed to lie on algebraic varieties. If $\mathbf{D}^{(a)} = \mathbf{0}_{N\times R}$, for example, the algebraic variety is then the set of solutions of a system of homogeneous polynomial equations.

Other examples

One of the first appearances of the technique can be found in [70], discussing the decomposition of the fourth-order cumulant[19] in the context of underdetermined ICA. The presented so-called rank-1 detection device was later applied to obtain an algebraic algorithm for a CPD of which only one factor matrix has full column rank [99]. Indeed, the CPD $\mathcal{X} = \llbracket \mathbf{G}, \mathbf{H}, \mathbf{B} \rrbracket$ with $\mathbf{G} \in \mathbb{K}^{I\times R}$ and $\mathbf{H} \in \mathbb{K}^{J\times R}$ can be seen as a unique matrix factorization problem $\left(\mathbf{X}_{(3)}\right)^{\mathrm{T}} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$ with $\mathbf{A} = \mathbf{H} \odot \mathbf{G}$. Ported to this framework, the following function is then used:

$$\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, i_1, i_2, j_1, j_2) = e^{(1)}_{i_1 j_1} e^{(2)}_{i_2 j_2} + e^{(1)}_{i_2 j_2} e^{(2)}_{i_1 j_1} - e^{(1)}_{i_1 j_2} e^{(2)}_{i_2 j_1} - e^{(1)}_{i_2 j_1} e^{(2)}_{i_1 j_2}, \tag{2.25}$$

in which $\mathbf{E}^{(1)}$ and $\mathbf{E}^{(2)}$ are segmented versions of $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$, respectively, with dimensions $I \times J$. The matrices obtained by evaluating $\underline{z}$ in $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \times \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ or $\{\mathbf{a}_1, \ldots, \mathbf{a}_K\} \times \{\mathbf{a}_1, \ldots, \mathbf{a}_K\}$ can then be stacked for all meaningful combinations of parameter values $1 \leq i_1 < i_2 \leq I$ and $1 \leq j_1 < j_2 \leq J$, yielding the tensors $\mathcal{Z}^{(x)}$ and $\mathcal{Z}^{(a)}$, respectively. It can be seen that the diagonals of the slices of $\mathcal{Z}^{(a)}$ are all zero, following from the fact that each column of $\mathbf{A}$ can be written as $\mathbf{a}_r = \mathbf{h}_r \odot \mathbf{g}_r$. Hence, the discussed procedure can be applied to recover $\mathbf{B}$ and subsequently $\mathbf{G}$ and $\mathbf{H}$.

Furthermore, in [346], a similar technique is used to obtain an algebraic algorithm for the multilinear rank-$(L_r, L_r, 1)$ decomposition. The ideas are generalized in [130] to allow the computation of a CPD of which none of the factor matrices has full column rank.

The separation of constant modulus signals as illustrated in Section 2.7.2 has recently been extended to multi-modulus signals (such as 16-QAM signals), yielding a set of coupled CPDs [122]. Separation techniques based on the same approach as ACMA have also appeared for discrete / finite alphabet [172], constant power [411], binary [383] and on/off signals [385].

In Section 2.6, we have explored techniques to separate (evaluated) exponential functions and rational functions, allowing the recovery of $\mathbf{B}$. These models can be ported to this framework as well, recovering the separation matrix $\mathbf{W}$. For

[19] See Section 2.7.3.

example, in the case of exponentials [339], $\mathcal{F}^{(x)}$ can then be constructed using the following function:

$$\underline{z}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, n) = 2u_n^{(1)}u_n^{(2)} - u_{n+1}^{(1)}u_{n-1}^{(2)} - u_{n-1}^{(1)}u_{n+1}^{(2)}$$

with $1 < n < N$. It is easy to see that the slice diagonals of $\mathcal{F}^{(a)}$ are zero if and only if the vectors $\mathbf{a}_r$ are exponentials, as they satisfy the property $a_{nr}^2 = a_{n-1,r}a_{n+1,r}$. Similar to the technique presented in Section 2.6.4, this technique is more flexible compared with the Hankelization and segmentation procedures and can be used if some samples are missing, if the signals are not equidistantly sampled, or for non-standard array shapes.

2.7.3 Direct construction of a tensor

Instead of constructing a tensor by concatenating matrices, it is also possible to directly obtain a higher-order tensor from a given matrix $\mathbf{X}$. Higher-order moments and cumulants, for example, are two well-known types of higher-order statistics [276]. Although defined as higher-order derivatives of characteristic functions, they are most common in their explicit form as we will point out. Moments are closely connected with N-grams from natural language processing [59]. We will illustrate the use of moments and N-grams for clustering in Section 2.7.3 and make links with stacked outer products. Cumulants, on the other hand, are prevalent in independent component analysis as will be shown in Section 2.7.3. The use of adjacency tensors in graph analysis is discussed in Section 2.7.3, while neural networks have also clear links with tensors through score functions. The latter is covered in Section 2.7.3.

Insert 18 — Unique matrix factorization: The approach for unique matrix factorization using the direct construction of a higher-order tensor typically consists of the following. Let the tensorization technique be defined by the function $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$, and let $\mathcal{Z}^{(x)}$ and $\mathcal{Z}^{(a)}$ be the order-D tensorized versions of $\mathbf{X}$ and $\mathbf{A}$, respectively. Similar to the discussion in Insert 15, a multilinear function $\underline{z}$ allows one to write

$$\mathcal{Z}^{(x)} = \mathcal{Z}^{(a)} \bullet_1 \mathbf{B} \cdots \bullet_D \mathbf{B}.$$

Under the assumption of diagonality of $\mathcal{Z}^{(a)}$, $\mathcal{Z}^{(x)}$ admits the CPD $\mathcal{Z}^{(x)} = \left[\!\left[\mathbf{d}^{(a)}; \mathbf{B}, \ldots, \mathbf{B}\right]\!\right]$ with $d_r^{(a)} = z_{r\cdots r}^{(a)}$. This CPD is unique under mild conditions such that $\mathbf{B}$ (and subsequently $\mathbf{A}$) can be recovered.

Higher-order moments and N-grams

Higher-order statistics (HOS) have seen a broad use in statistical theory, signal processing and data analysis [89, 266, 277]. The original definition of HOS is based on higher-order derivatives of a characteristic function, while a better known version of the definition is based on the evaluation of a function $\underline{z}$ in a product set. We will show how these two versions are linked to each other. In this section, we will limit the discussion to higher-order moments; higher-order cumulants are covered in Section 2.7.3. The concepts apply to covariance matrices (Section 2.7.1) as well, as covariance matrices are second-order versions of cumulants. Higher-order moments are also known as N-grams in natural language processing and computational linguistics as we will point out [59].

We consider again the set of stochastic variables χ_k instead of the observations $\mathbf{x}_k$. The derivations are also valid for the latter by substituting the expectancy operator with the mean. Let

$$\underline{g}(\upsilon) = \mathrm{E}\left\{e^{j\upsilon}\right\} \tag{2.26}$$

with υ a stochastic variable and let us substitute $\upsilon = \omega_1\chi_1 + \ldots + \omega_K\chi_k$ such that the function

$$\underline{h}(\omega_1, \ldots, \omega_K) = \mathrm{E}\left\{e^{j(\omega_1\chi_1 + \ldots + \omega_K\chi_K)}\right\} \tag{2.27}$$

is obtained. This function is also known as the first characteristic function. Moments of order D are now defined as the stacked derivatives of order D of $\underline{h}(\omega_1, \ldots, \omega_K)$ evaluated in $\omega_1, \ldots, \omega_K = 0$ and scaled by $1/j^D$. The stacking of higher-order derivatives has already been discussed in Section 2.4.3. Formally, we have the moment $\mathcal{M} \in \mathbb{R}^{K \times \cdots \times K}$ of order D with for all indices:

$$m_{k_1 \cdots k_D} = \frac{1}{j^D} \left. \frac{\partial^D \mathrm{E}\left\{e^{j(\omega_1\chi_1 + \cdots + \omega_K\chi_K)}\right\}}{\partial\omega_{k_1} \cdots \partial\omega_{k_D}} \right|_{\omega_1, \ldots, \omega_D = 0}. \tag{2.28}$$

Hence, each entry of $\mathcal{M}$ contains the derivative of order D $\underline{h}$ to a combination of D variables from $\omega_1, \ldots, \omega_K$ in which repetition is allowed. This is the classical derivative-based definition of the higher-order moment. However, in practice, it can be cumbersome to compute these higher-order derivatives. Another version of the definition is much more established, which can be linked to the evaluation of a function $\underline{z}(\upsilon_1, \ldots, \upsilon_D)$ in a product set.

Let us first compute the derivatives by applying the chain rule. Then, one can write

$$\mathcal{M} = \mathrm{E}\left\{ \left(\frac{1}{j^D} \left. \frac{\partial^D \underline{g}(\upsilon)}{\partial\upsilon^D} \right|_{\upsilon=0} \right) (\boldsymbol{\chi} \otimes \cdots \otimes \boldsymbol{\chi}) \right\}.$$

The scalar term in the brackets is equal to 1. Hence, the following simple well-known explicit formula for the higher-order moment emerges with for all indices:

$$m_{k_1 \cdots k_D} = \mathrm{E}\left\{\chi_{k_1} \cdots \chi_{k_D}\right\}. \tag{2.29}$$

The attentive reader has now been able to detect two different types of tensorization in both definitions of $\mathcal{M}$. On the one hand, following the definition in (2.28), the tensor $\mathcal{M}$ consists of higher-order derivatives of the function $\underline{g}(\upsilon)$. On the other hand, following the definition in (2.29), $\mathcal{M}$ consists of evaluations of the function

$$\underline{z}(\upsilon^{(1)}, \ldots, \upsilon^{(D)}) = \mathrm{E}\left\{\upsilon^{(1)} \cdots \upsilon^{(D)}\right\} \tag{2.30}$$

in the product set $\{\chi_1, \ldots, \chi_K\} \times \cdots \times \{\chi_1, \ldots, \chi_K\}$. The two tensorization techniques are associated with each other because of the use of the linear combination substitution in (2.27). (2.30) can be seen as a simplified form of

$$\underline{z}(\upsilon^{(1)}, \ldots, \upsilon^{(D)}) = \frac{1}{j^D} \frac{\partial^D \mathrm{E}\left\{e^{j(\omega_1 \upsilon^{(1)} + \cdots + \omega_D \upsilon^{(D)})}\right\}}{\partial \omega_1 \cdots \partial \omega_D}\bigg|_{\omega_d = 0}.$$

The latter is the polar form of $\underline{g}(\upsilon)$ obtained through polarization [233, 355]. The two different derivations leading to the same tensor $\mathcal{M}$ can then be summarized as follows:

1. Substitute υ in $\underline{g}(\upsilon)$ with $\omega_1\chi_1 + \cdots + \omega_K\chi_K$ and compute the higher-order derivatives of $\underline{h}$ in zero for each combination of D variables ω_k or
2. Substitute υ in $\underline{g}(\upsilon)$ with $\omega_1\upsilon^{(1)} + \cdots + \omega_D\upsilon^{(D)}$, compute the higher-order derivative to $\omega_1, \ldots, \omega_D$ in zero and evaluate $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ in the Cartesian product set $\{\chi_1, \ldots, \chi_K\} \times \cdots \times \{\chi_1, \ldots, \chi_K\}$.

The same tensor emerges by following both procedures.

In practice, often only observations of the stochastic variables are available. The expressions remain valid by substituting υ and χ with $\mathbf{u}$ and $\mathbf{x}$, respectively, and by replacing the expectancy operator by the mean. Procedures 1 and 2 in this discrete case are then visualized in Figure 2.14, respectively on the right and left.

Although used in a different context, N-grams are mathematically equal to higher-order moments. N-grams are defined as the co-occurrence probabilities of objects, such as words in the same document or topic. By collecting the occurrence probabilities of words in each document or topic in a matrix $\mathbf{X}$, the N-grams are equal to the order-N moments of $\mathbf{X}$.

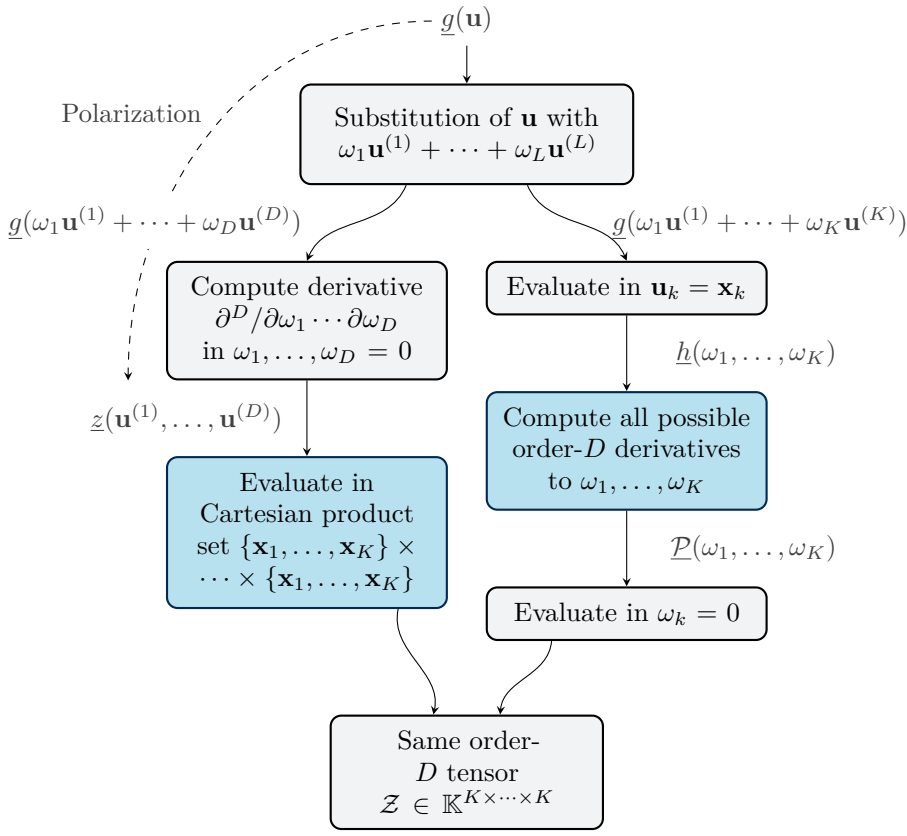


Figure 2.14: Illustration of the order-D tensorization of K data vectors $\mathbf{x}_1,\ldots,\mathbf{x}_K \in \mathbb{K}^N$ based on a scalar function $\underline{g}(\mathbf{u})$ with $\mathbf{u} \in \mathbb{K}^N$. Two different sequences of steps yield the same tensor $\mathcal{Z}$, each induced by fundamentally different tensorization techniques (in blue). The functions $\underline{g}(\upsilon) = \mathrm{E}\left\{e^{j\upsilon}\right\}$ and $\underline{g}(\upsilon) = \log \mathrm{E}\left\{e^{j\upsilon}\right\}$ then lead to the higher-order moment $\mathcal{M}$ and higher-order cumulant $\mathcal{C}$, respectively, as discussed in Sections 2.7.3 and 2.7.3.

Insert 19 — Clustering: Let us recall the clustering problem from Insert 16. By computing, e.g., the third-order moment (or trigram) of $\mathbf{X}$, we obtain a CPD:

$$\mathcal{M}^{(x)} = \mathcal{M}^{(a)} \cdot_1 \mathbf{B} \cdot_2 \mathbf{B} \cdot_3 \mathbf{B} = \left[\!\left[\boldsymbol{\pi}^{(a)}; \mathbf{B}, \mathbf{B}, \mathbf{B} \right]\!\right], \tag{2.31}$$

with $\boldsymbol{\pi}^{(a)}$ the diagonal of $\mathcal{M}^{(a)}$. Indeed, if $\mathbf{A}$ is a selection matrix, $\mathcal{M}^{(a)}$ is diagonal. The vector $\boldsymbol{\pi}^{(a)}$ then contains the probabilities of each cluster (or topic in topic modeling) [18, 193, 324].

It can be seen that the higher-order moments are reduced versions of the stacked higher-order outer products in Section 2.7.1. A third-order moment, then, is obtained by applying a summation along the fourth mode of the tensor obtained by stacking third-order outer products. The CPD in (2.31) is then a direct consequence from the CPD obtained in (2.20).

In the presence of noise, some additional measures can be taken. For example, the smallest eigenvalues of the covariance matrix of $\mathbf{X}$ give an indication of the noise variance [193]. By subtracting the noise contributions from the moments, the noise can be neutralized to some extent. It is not straightforward to incorporate the knowledge of the noise statistics in stacked outer products from Section 2.7.1.

Higher-order cumulants

In the previous section, we have explained that higher-order moments can be defined as the higher-order derivatives of the first characteristic function, and that they can also be defined as the evaluation of some multivariate function in a product set. The same is true for higher-order cumulants, which follow from the function $\underline{g}(\upsilon) = \log \mathrm{E}\left\{e^{j\upsilon}\right\}$. Note the additional log compared to (2.26). After substitution of $\omega_1\upsilon_1 + \cdots + \omega_K\upsilon_K$, the second characteristic function is obtained:

$$\underline{h}(\omega_1, \ldots, \omega_K) = \log \mathrm{E}\left\{ e^{j(\omega_1\chi_1 + \ldots + \omega_K\chi_K)} \right\}.$$

The cumulant of order D [39, 354] is defined as the stacking of all derivatives of order D of $\underline{h}(\omega_1, \ldots, \omega_K)$ evaluated in the origin. The second-order cumulant which is known as the covariance matrix consists of the entries $c_{k_1k_2} = \mathrm{E}\left\{\tilde{\chi}_{k_1}\tilde{\chi}_{k_2}\right\}$ with $\tilde{\chi}$ the zero-mean version of χ, while the entries of the

fourth-order cumulant are defined as

$$c_{k_1\cdots k_4} = \mathrm{E}\{\tilde{\chi}_{k_1}\tilde{\chi}_{k_2}\tilde{\chi}_{k_3}\tilde{\chi}_{k_4}\} - \mathrm{E}\{\tilde{\chi}_{k_1}\tilde{\chi}_{k_2}\}\mathrm{E}\{\tilde{\chi}_{k_3}\tilde{\chi}_{k_4}\} - \mathrm{E}\{\tilde{\chi}_{k_1}\tilde{\chi}_{k_3}\}\mathrm{E}\{\tilde{\chi}_{k_2}\tilde{\chi}_{k_4}\} - \mathrm{E}\{\tilde{\chi}_{k_1}\tilde{\chi}_{k_4}\}\mathrm{E}\{\tilde{\chi}_{k_2}\tilde{\chi}_{k_3}\}.$$

Alternatively, $\mathcal{C}$ can be defined as the evaluation of the polarized form $\underline{z}$ of $\underline{g}(v)$ in the Cartesian product set with

$$\underline{z}(v_1,\ldots,v_4) = \mathrm{E}\{\tilde{v}_1\tilde{v}_2\tilde{v}_3\tilde{v}_4\} - \mathrm{E}\{\tilde{v}_1\tilde{v}_2\}\mathrm{E}\{\tilde{v}_3\tilde{v}_4\} - \mathrm{E}\{\tilde{v}_1\tilde{v}_3\}\mathrm{E}\{\tilde{v}_2\tilde{v}_4\} - \mathrm{E}\{\tilde{v}_1\tilde{v}_4\}\mathrm{E}\{\tilde{v}_2\tilde{v}_3\}. \tag{2.32}$$

It is well-known that mutually independent stochastic variables $\chi_1,\ldots,\chi_K$ yield a diagonal cumulant [266, 267], but it is less clear why. In contrast to moments, it can be seen that the diagonality emerges because of the presence of the log function. Relying on the fact that $\mathrm{E}\{v_1v_2\} = \mathrm{E}\{v_1\}\mathrm{E}\{v_2\}$ for mutually independent stochastic variables v_1 and v_2, $\underline{h}(\omega_1,\ldots,\omega_K)$ can be written as a sum of univariate functions with

$$\underline{h}(\omega_1,\ldots,\omega_K) = \log\mathrm{E}\left\{e^{j\omega_1\chi_1}\right\} + \cdots + \log\mathrm{E}\left\{e^{j\omega_K\chi_K}\right\}.$$

We have seen in Section 2.4.3 that a sum of univariate functions yields a diagonal higher-order derivative tensor.

Again, in the discrete case, the expectancy operator, v and χ can be replaced by the mean[20], $\mathbf{u}$ and $\mathbf{x}$, respectively. The obtained sample cumulant $\hat{\mathcal{C}}$ will be approximately[21] diagonal for mutually independent signals.

Insert 20 — Independent component analysis: Because of the diagonality of cumulant tensors for mutually independent stochastic variables, cumulants are often used for the separation of non-Gaussian mutually independent signals in ICA [89]. This readily yields a CPD, as discussed in Insert 18. As the covariance matrix does not provide sufficient information, and as the third-order cumulant tensor is zero for symmetric distributions, fourth-order cumulants are most common. In early ICA algorithms, the covariance matrix was used in a PCA-based preprocessing step to obtain orthogonal factor matrices in the subsequent tensor decomposition [69, 71]. In the presence of (Gaussian) noise, however, it has been shown that the PCA

[20] Better estimators for the cumulant entries exist such as the k-statistics which are the minimum-variance unbiased estimators [213]

[21] Observe the difference with the diagonal property of higher-order moments. While a finite sample cumulant is only approximately diagonal for mutually independent variables, the higher-order moment is exactly diagonal for selection matrices.

step can introduce significant errors and prevents asymptotically unbiased estimates [110]. Furthermore, developments have led to well-established CPD algorithms that are able to cope well with non-orthogonal and possibly ill-conditioned factor matrices.

Rather than using only fourth-order cumulants [70], it can still be beneficial to combine these cumulants with covariance matrices (and other meaningful lower-order statistics) which typically suffer less from estimation errors [49, 153]. For example, the covariance matrix can be concatenated with the reshaped version of the fourth-order cumulant of size $K \times K \times K^2$, whereas coupled tensor decompositions might provide a more flexible approach. Second, it can be beneficial to assume the covariance matrix diagonal as unknown, as it consists of the largest noise contribution in the case of uncorrelated noise components. Note that a series of cumulants up to order D can also be stacked in a $(K+1) \times \cdots \times (K+1)$ tensor of order D and subsequently decomposed as explained in Section 2.4.3.

In any case, sufficient importance should be placed on the weighting of the various statistics. A statistically optimal weighting exists, but can be hard to find. We refer the interested leader to [357, 404], while weighting techniques dealing with stacked covariance matrices from Section 2.7.1 can be found in [68, 144, 403].

Graph adjacency tensors

Graphs are defined as sets of vertices or nodes and edges, the latter typically expressing relations between those nodes. Nodes can represent objects, persons, locations or events, amongst others. Graphs are often seen as abstractions of networks (such as social, computer or supply networks), and can be (un)weighted and/or (un)directed [356]. Although typically visualized graphically, cf. Figure 2.15, one can express a graph with K nodes uniquely with its square $K \times K$ adjacency matrix $\mathbf{X}$, which will be binary and symmetric for unweighted and undirected graphs, respectively. $\mathbf{X}$ contains the second-order structure of the graph, i.e., entry x_{ik} expresses the connection from node i to node k with $x_{ik} = 0$ denoting a non-existing connection. Hence, porting this concept to the framework in this section, a graph corresponds to a given set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K$, with the kth vector denoting the connections to node k.

Tensors have appeared in graph theory before, typically in the form of stacked adjacency matrices of different graph instances. Each adjacency matrix then corresponds to the same set of nodes but to a different set of edges, typically forming a layered or time-evolving graph. For example, airports can be linked

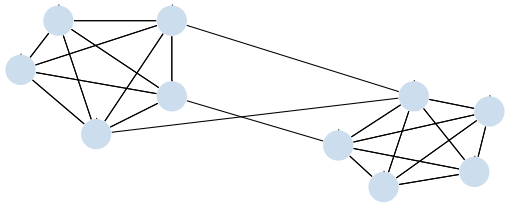


Figure 2.15: An example of a graph with two clearly identifiable yet connected clusters.

through different airlines [400], websites can be linked through a number of different keywords [224], or authors can be linked through different citations or collaborations [292]. In [319, 320], adjacency matrices corresponding to different time instances are stacked. The tensorization technique is then similar to the concept discussed in Section 2.7.1, consisting of the stacking of different (parameter-dependent) matrices.

On the other hand, tensors appear more naturally by considering higher-order graph structures, which can be entirely deduced from the adjacency matrix. $\mathbf{X}$ can then be tensorized to the adjacency tensor $\mathcal{Z} \in \mathbb{K}^{K \times K \times K}$, using suitable definitions. For example, for a binary symmetric $\mathbf{X}$, the triangle adjacency tensor consists of entries $z_{ijk} = 1$ if $x_{ij} = x_{ik} = x_{jk} = 1$ or if any two indices are equal and $z_{ijk} = 0$ otherwise, i.e., 'triangle' connections between nodes i, j and k are extracted [252, 374]. Other higher-order types such as stars correspond to other adjacency tensor definitions [374].

Adjacency tensors also form the basis of transition tensors corresponding to higher-order Markov chains. For example, the entries p_{ijk} of a transition tensor $\mathcal{P}$ indicate the transition probability from current state j to state i given the previous state k. Transition tensors have been used in applications such as multilinear pagerank [41, 163, 400].

Insert 21 — Graph clustering: A common objective in graph theory is the identification of specific clusters in graphs, i.e., well-connected subgraphs. Let us consider a graph with K nodes and R clusters, with the rth cluster containing K_r nodes. Nodes in the same cluster are assumed to be fully connected and there exist no connections between clusters. For $R = 2$, a so-called bipartite graph is obtained.

Let us construct the triangle tensor $\mathcal{Z}$. By ordering the nodes of the graph such that nodes of the same cluster are grouped, it can be seen that $\mathcal{Z}$ is a block diagonal tensor with blocks consisting of all ones. Note that the tensor with all ones has rank 1 and can be written as $\mathbf{1}_K \otimes \mathbf{1}_K \otimes \mathbf{1}_K$. It is easy to

show that $\mathcal{Z}$ admits a rank-R CPD $\mathcal{Z} = [\![\mathbf{U}, \mathbf{U}, \mathbf{U}]\!]$ with factor matrix [17, 41, 374]

$$\mathbf{U} = \begin{bmatrix} \mathbf{1}_{I_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{I_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{I_R} \end{bmatrix}.$$

Even if the nodes are not reordered, the clusters can be easily extracted using a truncation procedure or data clustering techniques such as k-means [185]. In the non-ideal case where clusters are not fully connected and with connections between clusters, such as in Figure 2.15, the CPD holds only approximately, but still a suboptimal estimate of $\mathbf{U}$ can be obtained [374]. The clusters can then easily be extracted.

Higher-order score functions

Higher-order score functions are similar to moments [205]. Given a stochastic vector $\boldsymbol{\chi} \in \mathbb{K}^K$ and its probability density function $\underline{p}(\mathbf{x})$ in function of an observation $\mathbf{x} \in \mathbb{K}^K$, the order-$D$ score function $\underline{\mathcal{S}}(\mathbf{x}) \in \mathbb{K}^{K \times \cdots \times K}$ is defined as

$$\underline{\mathcal{S}}(\mathbf{x}) = (-1)^D \frac{\underline{\mathcal{P}}(\mathbf{x})}{\underline{p}(\mathbf{x})}, \tag{2.33}$$

with $\underline{\mathcal{P}}$ holding the higher-order derivatives of $\underline{p}$ as discussed in Section 2.4.3. Considering a particular model which maps an observation $\mathbf{x}$ of $\boldsymbol{\chi}$ to $\underline{y}(\mathbf{x})$, the so-called corresponding cross-moment tensor $\mathcal{T} \in \mathbb{K}^{K \times \cdots \times K}$ of order D is defined as

$$\mathcal{T} = \mathrm{E}\left\{\underline{y}(\mathbf{x})\underline{\mathcal{S}}(\mathbf{x})\right\}. \tag{2.34}$$

The cross-moment tensor has special properties for specific types of models, e.g., for neural networks with a single hidden layer as we discuss in Insert 22

Insert 22 — Neural networks: Let us consider a given set of feature vectors with corresponding scalar labels. We will model the classifier by a neural network with a single hidden layer.

Note that the results can easily be generalized to vector labels. The neural network is modeled as follows:

$$\underline{y}(\mathbf{x}) = \mathbf{b}^{\mathrm{T}} \underline{\sigma}(\mathbf{A}^{\mathrm{T}} \mathbf{x}), \tag{2.35}$$

as depicted in Figure 2.16. The so-called activation function $\underline{\sigma}$ is evaluated element-wise and is assumed to be sufficiently differentiable. Note that constant terms can be added by considering an additional input $x_{K+1} = 1$, cf. the homogenization procedure from Section 2.5.2.

Let us assume N feature vectors are given which are the rows of the given matrix $\mathbf{X}$. The vector $\mathbf{x}$ in (2.33), (2.34) and (2.35) can then be substituted with the nth row of $\mathbf{X}$. Each feature vector can be seen as an observation of a stochastic vector $\boldsymbol{\chi}$, of which we know the probability density function $\underline{p}$. The nth feature vector corresponds with the label y_n. The goal consists of finding the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ given these observations. Typically, these are recovered by training the neural networks using methods such as back propagation [186].

Alternatively, let us construct the cross-moment tensor $\mathcal{T}$. In practice, we have

$$\mathcal{T} = \frac{1}{N} \sum_n y_n \underline{S}((\mathbf{X}^{\mathrm{T}})_n).$$

If the labels y_n satisfy the model in (2.35), it has been proven that the cross-moment tensor $\mathcal{T}$ of order D from (2.34) admits a CPD $\mathcal{T} = [\![\mathbf{A}, \ldots, \mathbf{A}]\!]$ [204, 205]. By constructing $\mathcal{T}$ and applying a CPD, the matrix $\mathbf{A}$ can readily be obtained. Hence, computing the CPD is equivalent to training the neural network. Note that algebraic CPD algorithms exist which do not suffer from local optima, unlike optimization-based algorithms. If the labels do not perfectly follow the model, algebraic CPD algorithms can still provide suboptimal estimates of $\mathbf{A}$ which can serve as initializations for optimization-based techniques.

The appearance of the CPD can be understood from the fact that the function $\underline{\sigma}$ is applied element-wise, as is also the case in the decoupling of multivariate functions discussed in Insert 3. Indeed, it can be shown that the neural network model is a specific instance of the model in (2.4). However, the technique from Insert 3 cannot be applied in the case of neural networks as the higher-order derivatives of $\underline{y}$ are typically unknown. The method based on score functions considers the derivatives of the probability density function of the inputs rather than the derivatives of $\underline{y}$.

Further notes on polarization

Higher-order statistics, adjacency tensors and score functions might be the most well-known techniques that map a given matrix $\mathbf{X}$ to a multiway tensor. As

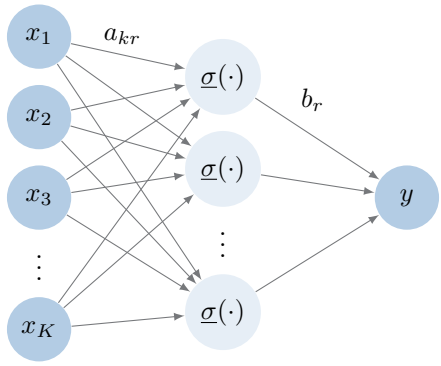


Figure 2.16: Visualization of a neural network with one hidden layer. The input is an observation of the stochastic vector $\boldsymbol{\chi} \in \mathbb{K}^K$. The neural network considers a scalar label y but this can be extended easily to a vector output.

discussed, each function $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ has the capability of tensorizing a given set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K$ to a tensor of order D. To be able to apply tensor tools in a subsequent step, the function $\underline{z}$ typically needs to be multilinear.

The polarization-based discussion in Section 2.7.3 provides a novel framework as it shows that each function $\underline{g}(\mathbf{u})$ leads to a function $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$, called the polar form of $\underline{g}(\mathbf{u})$, which can be used to map a given set of vectors to a tensor. Furthermore, it is well known in algebra that the polar form is multilinear [233, 355]. We reproduce a brief proof of this in the appendix of this chapter. This is not unimportant: it shows that a meaningful multilinear tensorization technique can be obtained by choosing a suitable function $\underline{g}(\mathbf{u})$ or $\underline{g}(\upsilon)$.

We have explained that both higher-order moments and higher-order cumulants originate from a particular choice of $\underline{g}(\upsilon)$, namely from $\underline{g}(\upsilon) = \mathrm{E}\left\{e^{j\upsilon}\right\}$ and $\underline{g}(\upsilon) = \log \mathrm{E}\left\{e^{j\upsilon}\right\}$, respectively. Although not presented in such a way, the determinant function $\underline{g}(\mathbf{U}) = \det(\mathbf{U})$ lies at the core of the algebraic algorithm developed in [130] to compute a CPD in which none of the factor matrices has full column rank. This function gives rise to the mixed discriminant function

$$\underline{z}(\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(D)}) = \sum_{d=1}^{D} \sum_{\substack{1 \leq i_d \leq D \\ i_1 < \cdots < i_D}} (-1)^{(D-d)} \det(\mathbf{U}^{(i_1)} + \cdots + \mathbf{U}^{(i_d)}). \quad (2.36)$$

More accurately, the vector function $\underline{\mathbf{g}}(\mathbf{U}) = \mathrm{vec}\left(\mathbf{C}_M(\mathbf{U})\right)$ is applied which leads to the (vectorized) polarized compound matrix function $\underline{\boldsymbol{z}}(\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(D)})$. The latter can be expressed as in (2.36) but with $\det(\cdot)$ replaced by $\mathrm{vec}\left(\mathbf{C}_D(\cdot)\right)$. It can be seen that, for $D = 2$, $\underline{\boldsymbol{z}}$ contains the scalar functions defined in

(2.25). Furthermore, it should not come as a surprise that (2.36) and (2.32) (the definition of the fourth-order cumulant) have a very similar structure, as they are both derived in a similar way from a function $\underline{g}$.

2.8 Further discussions and future work

A number of different concepts have been proposed on how a vector or matrix can be mapped to a tensor. A tensorization technique should not uniquely be identified with a single concept, as the concepts can also be joined. For example, spatio-temporal cumulants merge the concepts of parameter variation from Section 2.7.1 and the direct construction of a cumulant from Section 2.7.3. Hence, the spatio-temporal cumulants not only take into account the non-Gaussian character of the signals but also their non-stationarity. For example, the spatio-temporal generalization of the fourth-order cumulant yields a tensor of order seven, each entry depending on a combination of four signals and three time lags. Spatio-temporal cumulants in combination with tensor decompositions are particularly useful for blind system identification [150, 247, 373].

Furthermore, tensorization techniques can also be combined and applied simultaneously. Given a matrix $\mathbf{X}$, twofold tensorization applies a tensorization technique on each mode of $\mathbf{X}$ simultaneously. This can be used to incorporate assumptions on both $\mathbf{A}$ and $\mathbf{B}$ given the model $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$. This twofold tensorization then results in a tensor of order four or higher. For example, in the context of instantaneous BSS, twofold segmentation can be used to exploit the Kronecker structure of both $\mathbf{A}$ (the source level) and $\mathbf{B}$ (the mixing level) [51]. Similarly, if $\mathbf{B}$ has a Kronecker structure and $\mathbf{A}$ contains mutually independent and non-Gaussian signals, a combination of segmentation and higher-order cumulants can be used in a context of ICA [116]. Given a matrix $\mathbf{X}$, the construction of a compound matrix as discussed in Insert 2 can also be seen as twofold tensorization.

Many of the tensorization techniques have been illustrated based on a number of applications such as ICA, clustering and the training of neural networks. While relatively simple underlying models have been considered, advanced models might be more appropriate in a real-life setting. Rather than instantaneous ICA, single-membership cluster models or single-layer neural networks, one can consider convolutional ICA, mixed-membership cluster models, or neural networks with multiple hidden layers (deep neural networks) or with feedback (recurrent neural networks). While tensorization techniques have already seen a broad use in the convolutive generalization of ICA [150, 247, 373], one can imagine that the power of tensor tools can also be employed in the context of

the other generalizations. A limited number of results have already appeared, discussing the use of CPD in mixed-membership models [17] and the hierarchical Tucker decomposition for deep neural networks [88].

2.9 Conclusion

This chapter has covered a tensorization framework and has surveyed a large number of tensorization techniques which have been presented rather disparately in the literature. Tensorization is defined as the mapping of a vector or matrix to a tensor or, more generally, of mapping lower-order data to higher-order data. It has been illustrated how the techniques are fundamental to tensor-based methods for deterministic blind signal separation, independent component analysis, data clustering and topic modeling, graph clustering and the training of neural networks, among others. Links between different tensorization techniques have been established, such as between segmentation and time–frequency and time–scale representations, between higher-order statistics and mixed discriminants through a process called polarization, and between the analytical constant modulus algorithm and other algebraic variety-based methods. This polarization procedure allows the development of new multilinear tensorization techniques. Furthermore, we have recognized that a number of problems that are ubiquitous in engineering can be well represented by tensors. For example, the approximation of a non-linear function using a power series expansion is equivalent to representing the non-linear function by a tensor, and finding the global optimum of a polynomial within a finite interval is equivalent to finding the best symmetric rank-1 approximation of a tensor.

Acknowledgment

The authors would like to thank Ignat Domanov for the insights on algebraic geometry and polarization, Marc Van Barel for the discussions on Löwner tensors and Eleftherios Kofidis for the discussions on filter banks.

Constructing multilinear functions using polarization

Theorem 2.1. *Each possibly non-linear function* $\underline{g}(\mathbf{u})$ *with* $\mathbf{u} \in \mathbb{K}^N$ *of which the order-D derivatives exist in* $\mathbf{u} = \mathbf{0}$ *leads to a multilinear mapping* $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ *of the form*

$$\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)}) = \left. \frac{\partial^D \underline{g}(\omega_1 \mathbf{u}^{(1)} + \cdots + \omega_D \mathbf{u}^{(D)})}{\partial \omega_1 \cdots \partial \omega_D} \right|_{\omega_d = 0}.$$

Proof. By applying the chain rule, one can see that

$$\left. \frac{\partial \underline{g}(\omega_1 \mathbf{u}^{(1)} + \cdots + \omega_D \mathbf{u}^{(D)})}{\partial \omega_1} \right|_{\mathbf{0}} = \mathbf{u}^{(1)^{\mathrm{T}}} \nabla \underline{g}\big|_{\mathbf{0}}$$

with $\nabla \underline{g}$ the gradient of $\underline{g}(\mathbf{u})$. By applying the other $D-1$ derivatives, it follows that

$$\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)}) = \mathcal{P} \bullet_1 \mathbf{u}^{(1)^{\mathrm{T}}} \bullet_2 \mathbf{u}^{(2)^{\mathrm{T}}} \cdots \bullet_D \mathbf{u}^{(D)^{\mathrm{T}}}, \tag{2.37}$$

with $\mathcal{P}$ the order-D symmetric tensor of size $N \times \cdots \times N$ holding the higher-order derivatives of $\underline{g}(\mathbf{u})$ in $\mathbf{u} = \mathbf{0}$ as defined in (2.3). The function $\underline{z}$ from (2.37) is of the form in (2.6), i.e., it is a multilinear mapping. □

In algebra, typically, a polynomial function $\underline{g}(\mathbf{u})$ is considered. However, $\underline{g}(\mathbf{u})$ can be any non-linear function, as long as it is D times differentiable in the origin.

Corollary 2.1. *Each function* $\underline{g}(\mathbf{u})$ *may form the basis of a multilinear order-D tensorization technique. Given vectors* $\mathbf{x}_1, \ldots, \mathbf{x}_K$*, an order-D tensor of size* $K \times \cdots \times K$ *can be obtained by evaluating the corresponding polar form* $\underline{z}(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$ *in* $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \times \cdots \times \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$.

Chapter 3

Löwner-based blind signal separation of rational functions

Abstract A new blind signal separation (BSS) technique is proposed, enabling a deterministic separation of signals into rational functions. Rational functions can take on a wide range of forms, such as the well-known pole-like shape. The approach is a possible alternative for the well-known independent component analysis when the theoretical sources are not independent, such as for frequency spectra, or when only a small number of samples is available. The technique uses a low-rank decomposition on the tensorized version of the observed data matrix. The deterministic tensorization with Löwner matrices is comprehensively analyzed in this chapter. Uniqueness properties are investigated, and a connection with the separation into exponential polynomials is made. Finally, the technique is illustrated for fetal electrocardiogram extraction and with an application in the domain of fluorescence spectroscopy, enabling the identification of chemical analytes using only a single excitation-emission matrix.

Reference This chapter is a slightly adapted version of the article [124]. Changes are limited to layout and representation aspects. The candidate performed the research and wrote the article under the guidance of the coauthors.

3.1 Introduction

This chapter deals with the separation of linear mixtures of different source signals, known as blind signal separation (BSS). The general solution to this problem is not unique and various approaches have been proposed, ranging from applying independence assumptions to non-negativity and sparsity constraints [85]. The former has received the name independent component analysis (ICA) in which one assumes the sources to be statistically independent [89, 210]. ICA has already been applied in numerous applications in for example image processing, finance, telecommunications and biomedical sciences [64, 83, 89, 200]. It is a stochastic technique, tensorizing the observed matrix data using higher-order statistics [267]. Many applications do not involve stochastic and independent signals however, and one can also tensorize the data deterministically. In a source-related deterministic setting, as in this chapter, the separation of the observed signals is based on a specific source model.

A first possible source model could be the class of polynomials, as they have a simple form and well-understood properties. However, polynomial signal models often require a large number of coefficients. Second and more essential, polynomial models generally do not allow a unique separation, as will be illustrated in this chapter. With the family of exponential polynomials (sums and/or products of exponentials, sinusoids and/or polynomials), a source model is proposed in [101] which is applicable for blind signal separation by using a deterministic Hankel tensorization.

This chapter proposes the class of rational functions, contributing to a general framework of deterministic blind signal separation. They encompass the class of polynomials, and are able to model complicated structures with a fairly low degree in both the numerator and denominator; much like autoregressive–moving-average models are more powerful in the field of system identification than pure moving-average models [181]. Rational functions are mainly known because of their pole-like behavior (suitable when modeling frequency spectra, time-of-flight data, distribution functions, etc.) but can take on an extremely wide range of shapes in the complex domain; also smooth curves and signals with both low- and high-varying regions can be modeled. Furthermore, by considering uniqueness properties, we show in this chapter that rational functions are suitable in a BSS context. It is also illustrated that the sampling points need not be equidistant for the proposed technique, contrary to the Hankel technique from [101].

Another basis for BSS is sparse separation in which one uses a fixed signal dictionary with the weights optimized according to some sparse objectives [246, 408, 415]. The proposed technique in this chapter goes a step further as there

is no need for an initial signal dictionary: the dictionary itself is estimated too.

The assumption of rationality is implemented using the theory and properties of the Löwner matrix. This kind of matrix is well-known in the domain of system identification regarding rational interpolation [21, 23, 24], but is not acknowledged in other application domains; its definition is given in Section 3.2. The observed data matrix is tensorized using Löwner matrices, and the obtained tensor is analyzed using a block term tensor decomposition [84, 104, 113, 223]. Block component analysis (BCA) describes the use of block term decompositions to identify underlying components [102].

It is shown in this chapter that the solution with the assumption of rational sources is unique under mild conditions. This is important as it explains that the assumption is powerful and natural, while in the case of nonnegative constraints, for example, additional sparsity constraints need to be imposed to recover a unique solution [85]. Furthermore, techniques have been developed for ICA to recover more source signals than there are observed signals, i.e., for underdetermined mixtures [107, 151]. Because of the strong uniqueness results, this is readily extended in the separation method described. Simulations are presented in the final section of this chapter.

The technique was briefly described in [123]. We present the method with two illustrations using real-life datasets. The first illustration is antepartum fetal heart rate monitoring with the separation of mother and fetal electrocardiogram signals from multilead cutaneous potential recordings. An excellent separation is obtained, even for short sequences with coinciding heart beats. The second illustration covers the detection of chemical components in mixtures, using emission-excitation data from fluorescence spectroscopy. With the technique, only a single sample is sufficient to determine the concentrations and frequency spectra of the different chemical components.

We start by fixing the notation and discussing some basic definitions and decompositions in the field of multilinear algebra. In Section 3.2 the problem statement is examined and the Löwner-based technique is introduced. Section 3.3 contains a more advanced analysis. Uniqueness properties are considered in Section 3.4. A connection to the method from [101] is investigated in Section 3.5, and numerical experiments are performed in Section 3.6.

3.1.1 Notation and basic operations

Vectors (denoted by a bold, lowercase letter, e.g., $\mathbf{a}$) and matrices (denoted by a bold, uppercase letter, e.g., $\mathbf{A}$) can be generalized to higher orders, obtaining tensors. A general Nth order tensor of size $I_1 \times I_2 \times \cdots \times I_N$ is denoted by

a calligraphic letter as $\mathcal{A} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ (with $\mathbb{K}$ we mean $\mathbb{R}$ or $\mathbb{C}$). $\mathcal{A}$ is a multidimensional array with numerical values $a_{i_1 i_2 \cdots i_N} = \mathcal{A}(i_1, i_2, \ldots, i_N)$. The mode-$n$ vectors of $\mathcal{A}$ are constructed by fixing all but one index, e.g., $\mathbf{a} = \mathcal{A}(i_1, \ldots, i_{n-1}, :, i_{n+1}, \ldots, i_N)$.

A number of products can be defined in the domain of tensors. The mode-n tensor–matrix product between a tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ and a matrix $\mathbf{B} \in \mathbb{K}^{J \times I_n}$ is defined as

$$(\mathcal{A} \cdot_n \mathbf{B})_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum\nolimits_{i_n=1}^{I_n} a_{i_1 i_2 \cdots i_N} b_{j i_n}.$$

The outer product of two tensors $\mathcal{A} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathcal{B} \in \mathbb{K}^{J_1 \times J_2 \times \cdots \times J_M}$ is given as

$$(\mathcal{A} \otimes \mathcal{B})_{i_1 i_2 \cdots i_N j_1 j_2 \cdots j_M} = a_{i_1 i_2 \cdots i_N} b_{j_1 j_2 \cdots j_M}.$$

The matrices $\mathbf{A}^{\mathrm{T}}$ and $\mathbf{A}^{\dagger}$ denote the transpose and Moore-Penrose pseudo-inverse of $\mathbf{A}$, respectively. The Frobenius norm of a tensor is defined as the root of the sum of the squares of the elements: $\|\mathcal{A}\| = (\sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} |a_{i_1 \cdots i_N}|^2)^{1/2}$.

3.1.2 Rank definitions and basic tensor decompositions

The column (row) rank of a matrix $\mathbf{A}$ is the maximum number of linearly independent columns (rows) of $\mathbf{A}$. Note that the column rank is equal to the row rank for a matrix. We make a distinction between the rank and the multilinear rank.

First, a tensor $\mathcal{T}$ has rank 1 if it can be written as the outer product of some nonzero vectors: $\mathcal{T} = \mathbf{a}^{(1)} \otimes \mathbf{a}^{(2)} \otimes \ldots \otimes \mathbf{a}^{(N)}$. If one writes a tensor $\mathcal{T}$ as a linear combination of R rank-1 tensors, one obtains a polyadic decomposition (PD):

$$\begin{aligned} \mathcal{T} &= \sum\nolimits_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \cdots \otimes \mathbf{a}_r^{(N)} \\ &\triangleq \left[\!\!\left[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)} \right]\!\!\right]. \end{aligned}$$

If this R is minimal, the rank of $\mathcal{T}$ is defined as R, denoted by $r(\mathcal{T})$. The decomposition then becomes canonical (CPD).

Second, the mode-n rank of a tensor $\mathcal{T}$ is the dimension of the subspace spanned by its mode-n vectors. It is equal to the rank of the matrix constructed by stacking all the mode-n vectors one after the other. If the mode-1 rank, mode-2 rank and mode-3 rank of a third-order tensor are equal to L, M and

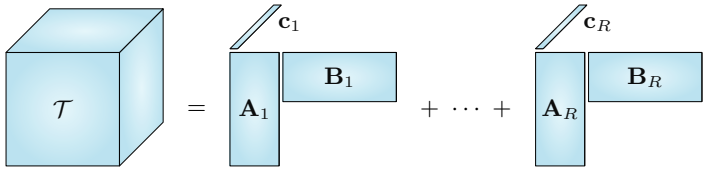


Figure 3.1: Decomposition of a tensor $\mathcal{T}$ in multilinear rank-$(L_r, L_r, 1)$ terms.

N, respectively, it is said to have trilinear rank (L, M, N). This becomes the multilinear rank when generalized to arbitrary order, obtaining the N-tuple $(R_1, R_2, \ldots, R_N)$. Connected to this multilinear rank is the Tucker decomposition:

$$\mathcal{T} = \mathcal{G} \cdot_1 \mathbf{A}^{(1)} \cdot_2 \mathbf{A}^{(2)} \cdots \cdot_N \mathbf{A}^{(N)} \triangleq \left[\!\left[\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)} \right]\!\right]$$

with $\mathcal{G} \in \mathbb{K}^{R_1 \times R_2 \times \ldots \times R_N}$ being a (typically small) core tensor. Related are the multilinear singular value decomposition and the low multilinear rank approximation; for details we refer to [105, 109, 112, 391].

In this chapter we make use of the block term decomposition (BTD), which starts from the idea of linearly combining tensors of low multilinear rank [104, 113]. For third-order tensors, one obtains the BTD in R rank-(L_r, M_r, N_r) terms. A special instance is the decomposition of a tensor $\mathcal{T} \in \mathbb{K}^{I_1 \times I_2 \times I_3}$ in rank-$(L_r, L_r, 1)$ terms (with $M_r = L_r$ and $N_r = 1$, for $1 \leq r \leq R$):

$$\mathcal{T} = \sum\nolimits_{r=1}^{R} \mathbf{E}_r \otimes \mathbf{c}_r, \tag{3.1}$$

with the matrix $\mathbf{E}_r \in \mathbb{K}^{I_1 \times I_2}$ having rank L_r and vector $\mathbf{c}_r \in \mathbb{K}^{I_3}$ being nonzero. Each matrix $\mathbf{E}_r$ can be factorized to give

$$\mathcal{T} = \sum\nolimits_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^{\mathrm{T}}) \otimes \mathbf{c}_r, \tag{3.2}$$

with $\mathbf{A}_r \in \mathbb{K}^{I_1 \times L_r}$, $\mathbf{B}_r \in \mathbb{K}^{I_2 \times L_r}$. It is visualized in Fig. 3.1.

3.2 Löwner-based blind signal separation

In this section, the technique for BSS with Löwner matrices is discussed and the use of the previously defined BTD in rank-$(L_r, L_r, 1)$ terms is explained. Section 3.2.1 discusses the model setup and Section 3.2.2 introduces Löwner matrices. Section 3.2.3 explains the tensorization technique. Finally, two different ways to recover the original sources are discussed.

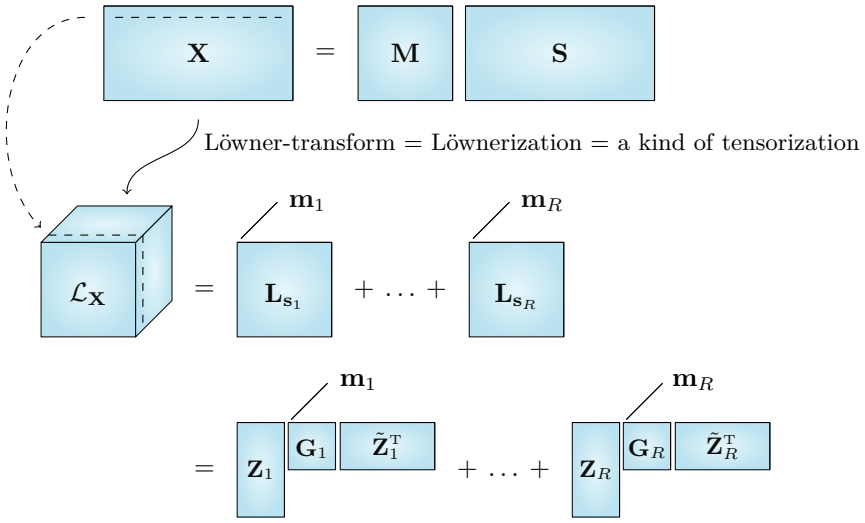


Figure 3.2: The observed data matrix is tensorized to stacked Löwner matrices, which are decomposed with a block term d ecomposition in rank-$(L_r, L_r, 1)$ terms. The mixing vectors $\mathbf{m}_1, \ldots, \mathbf{m}_R$ appear as factor vectors in the third mode, and the Löwner matrices of the sources appear in the first and second mode. The factor matrices have an interesting structure, explained in Section 3.3.

3.2.1 The blind signal separation problem

Assume we have R source signals being linearly mixed into K observed signals. For each signal N samples are available. We consider the following data model in BSS:

$$\mathbf{X} = \mathbf{MS} + \mathbf{N},$$

with $\mathbf{X} \in \mathbb{K}^{K \times N}$ containing the observed data, $\mathbf{S} \in \mathbb{K}^{R \times N}$ the R unknown source signals, $\mathbf{M} \in \mathbb{K}^{K \times R}$ the unknown mixing matrix and $\mathbf{N} \in \mathbb{K}^{K \times N}$ representing additive noise. The general goal in BSS is to recover the unknown sources in $\mathbf{S}$ and the unknown mixing vectors in $\mathbf{M}$, given only the observed data $\mathbf{X}$. We investigate the behavior related to added Gaussian noise in the experiments of Section 3.6 but omit $\mathbf{N}$ in the next analyses for convenience.

Broadly speaking, we will map each observed signal (each row in $\mathbf{X}$) to a Löwner matrix. By stacking these Löwner matrices, one obtains a tensor which is of low multilinear rank because of the working hypothesis, i.e., the source signals can be modeled by rational functions of low degrees. This hypothesis is satisfied in many applications. By decomposing the tensorized data, one can immediately

identify the mixing vectors. The reconstruction of the sources follows. Fig. 3.2 gives a comprehensive overview of the technique.

3.2.2 Löwner matrices

We first define the Löwner matrix for a function sampled in a point set T consisting of N distinct points:

Definition 3.1. *Given a function $f(t)$ sampled on points $T = \{t_1, t_2, \ldots, t_N\}$. We partition the point set T into two distinct point sets $X = \{x_1, x_2, \ldots, x_I\}$ and $Y = \{y_1, y_2, \ldots, y_J\}$ with $I + J = N$, and define the elements of the Löwner matrix $\mathbf{L} \in \mathbb{K}^{I \times J}$ as*

$$L_{i,j} = \frac{f(x_i) - f(y_j)}{x_i - y_j} \qquad \forall i, j.$$

We thus obtain the following matrix:

$$\mathbf{L} = \begin{bmatrix} \frac{f(x_1)-f(y_1)}{x_1-y_1} & \frac{f(x_1)-f(y_2)}{x_1-y_2} & \cdots & \frac{f(x_1)-f(y_J)}{x_1-y_J} \\ \frac{f(x_2)-f(y_1)}{x_2-y_1} & \frac{f(x_2)-f(y_2)}{x_2-y_2} & \cdots & \frac{f(x_2)-f(y_J)}{x_2-y_J} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{f(x_I)-f(y_1)}{x_I-y_1} & \frac{f(x_I)-f(y_2)}{x_I-y_2} & \cdots & \frac{f(x_I)-f(y_J)}{x_I-y_J} \end{bmatrix}.$$

In the literature, a parameter α is often used with $I = \alpha$ and $J = N - \alpha$. Matrix $\mathbf{L}$ is square when N is even and $\alpha = N/2$. Unless denoted otherwise, we assume $I = \alpha = \lceil N/2 \rceil$.

A Löwner matrix can also be constructed for point sets with coinciding sample points, for which we refer to other literature [152, 381]. Notice that the Löwner matrix corresponding to a constant function becomes the zero matrix.

The Löwner matrix has interesting properties in connection to rational functions. Let the degree of an irreducible rational function be defined as the maximum of the degrees of the polynomial in its numerator and the polynomial in its denominator. The following has been proven in [21, 32, 259]:

Theorem 3.1. *Given a Löwner matrix $\mathbf{L}$ of size $I \times J$ constructed from a function $f(t)$ sampled in a point set $T = \{t_1, \ldots, t_N\}$ with $N = I + J$. If $f(t)$ is a rational function of degree δ and if $I, J \geq \delta$, then $\mathbf{L}$ has rank δ:*

$$\operatorname{rank}(\mathbf{L}) = \delta = \deg(f).$$

This theorem is easy to verify for simple rational functions of low degree. For example, $f(t) = \frac{c}{t-p}$ gives $L_{i,j} = -c \cdot \frac{1}{x_i - p} \cdot \frac{1}{y_j - p}$. The corresponding matrix

$\mathbf{L}$ is of rank 1, as it can be written as an outer product of two vectors. The next section gives an insight into the structure of the Löwner matrix for a more general rational function.

Note that Theorem 3.1 is valid for any point set partitioning: two straightforward partitionings are the interleaved partitioning with $X = \{t_1, t_3, \ldots\}$ and $Y = \{t_2, t_4, \ldots\}$ and the block partitioning with $X = \{t_1, \ldots, t_I\}$ and $Y = \{t_{I+1}, \ldots, t_N\}$.

3.2.3 Tensorization and block term decomposition

Consider again the BSS model with $\mathbf{X} = \mathbf{MS}$. Let us map each row of $\mathbf{X}$ to a Löwner matrix of size $I \times J$, and stack these matrices in the third dimension obtaining a tensor $\mathcal{L}_{\mathbf{X}}$ of size $I \times J \times K$. We call this transformation the *Löwnerization* of matrix $\mathbf{X}$. As both the Löwner transform and the BSS model are linear, we can write:

$$\mathcal{L}_{\mathbf{X}} = \sum\nolimits_{r=1}^{R} \mathbf{L}_{\mathbf{s}_r} \otimes \mathbf{m}_r, \tag{3.3}$$

with $\mathbf{L}_{\mathbf{s}_r}$ the Löwner matrix and $\mathbf{m}_r$ the mixing vector of source r. If source $\mathbf{s}_r$, i.e., each rth row of $\mathbf{S}$, can be modeled as a rational function of some (low) degree δ_r, the matrix $\mathbf{L}_{\mathbf{s}_r}$ will have a (low) rank δ_r provided there are enough samples for $I, J \geq \max_r \delta_r$. The matrix $\mathbf{L}_{\mathbf{s}_r}$ admits a factorization with some $\mathbf{A}_r \in \mathbb{K}^{I \times \delta_r}$ and $\mathbf{B}_r \in \mathbb{K}^{J \times \delta_r}$:

$$\mathcal{L}_{\mathbf{X}} = \sum\nolimits_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^{\mathrm{T}}) \otimes \mathbf{m}_r, \tag{3.4}$$

which is precisely the decomposition in rank-$(L_r, L_r, 1)$ terms from (3.2) with $\delta_r = L_r$. In Section 3.3 we look into the factorization of $\mathbf{L}_{\mathbf{s}_r} = \mathbf{A}_r \mathbf{B}_r^{\mathrm{T}} = \mathbf{Z}_r \mathbf{G}_r \tilde{\mathbf{Z}}_r^{\mathrm{T}}$. In Section 3.4 the uniqueness properties are investigated regarding the use of a BTD in rank-$(L_r, L_r, 1)$ terms.

3.2.4 Recovery of the mixing matrix and the sources

The columns of the mixing matrix appear as the factor vectors in the third mode of the decomposition. The source signals can be reconstructed in two main ways. First, the estimated matrix $\hat{\mathbf{M}}$ can be inverted to calculate $\hat{\mathbf{S}} = \hat{\mathbf{M}}^{\dagger}\mathbf{X}$. This is the most straightforward method, and the default method for ICA. However, it is not applicable for underdetermined mixtures (with fewer observed signals than sources) or when the mixing matrix does not have full column rank. The

columns of $\hat{\mathbf{S}}$ are then determined up to arbitrary vectors in the null space of $\hat{\mathbf{M}}$, as in underdetermined ICA [107, 151].

A more general but more cumbersome technique is to recover the sources from the separated Löwner matrices in (3.3) and (3.4). Each recovered matrix $\mathbf{E}_r = \mathbf{A}_r \mathbf{B}_r^{\mathrm{T}}$ contains information about the corresponding source in a finite difference format. A linear system can be constructed to recover the source signal from their corresponding Löwner matrices:

$$\mathbf{s}_r = \arg\min_{\mathbf{s}_r} \frac{1}{2} \left\| \mathbf{F}\mathbf{s}_r - \mathrm{vec}(\hat{\mathbf{L}}_{\mathbf{s}_r}) \right\|^2 \quad \text{for } 1 \leq r \leq R,$$

with the matrix $\mathbf{F}$ being the reshaped matrix version of the following tensor $\mathcal{F}$ with size $I \times J \times N$:

$$\forall i, j, n: \; f_{i,j,n} = \begin{cases} \frac{1}{x_i - y_j} & \text{if } n = \phi(i), \\ \frac{-1}{x_i - y_j} & \text{if } n = \theta(j), \\ 0 & \text{elsewhere,} \end{cases}$$

with $\phi : i \to \{n \in \{1, \ldots, N\} : t_n = x_i\}$ and $\theta : j \to \{n \in \{1, \ldots, N\} : t_n = y_j\}$. $\mathbf{F}$ is constructed by vectorizing the third-order slices of $\mathcal{F}$ and stacking them as columns. With the point set $T = \{x_1, y_1, x_2, y_2\}$ for example, we have $\phi(1) = 1$, $\phi(2) = 3$, $\theta(1) = 2$ and $\theta(2) = 4$, and one obtains the following linear system:

$$\begin{bmatrix} (\hat{\mathbf{L}}_r)_{1,1} \\ (\hat{\mathbf{L}}_r)_{2,1} \\ (\hat{\mathbf{L}}_r)_{1,2} \\ (\hat{\mathbf{L}}_r)_{2,2} \end{bmatrix} = \begin{bmatrix} \frac{1}{x_1 - y_1} & \frac{-1}{x_1 - y_1} & 0 & 0 \\ 0 & \frac{-1}{x_2 - y_1} & \frac{1}{x_2 - y_1} & 0 \\ \frac{1}{x_1 - y_2} & 0 & 0 & \frac{-1}{x_1 - y_2} \\ 0 & 0 & \frac{1}{x_2 - y_2} & \frac{-1}{x_2 - y_2} \end{bmatrix} \begin{bmatrix} s_r(x_1) \\ s_r(y_1) \\ s_r(x_2) \\ s_r(y_2) \end{bmatrix}.$$

Let now μ_r be the DC component of the rth source. Observe that the vector $[\mu_r, \ldots, \mu_r]^{\mathrm{T}} \in \mathbb{K}^N$ is an element of the null space of the matrix $\mathbf{F}$. The vector $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_R]^{\mathrm{T}} \in \mathbb{K}^R$ can be found solving an additional linear system:

$$\boldsymbol{\mu} = \arg\min_{\boldsymbol{\mu}} \frac{1}{2} \left\| \mathbf{X} - \hat{\mathbf{M}} \left(\hat{\mathbf{S}} + \boldsymbol{\mu}\mathbf{e}^{\mathrm{T}} \right) \right\|^2 = \frac{1}{N} \left(\hat{\mathbf{M}}^{\dagger}\mathbf{X} - \hat{\mathbf{S}} \right) \mathbf{e}$$

with $\mathbf{e} = [1, \ldots, 1]^{\mathrm{T}} \in \mathbb{R}^N$. The vector $\boldsymbol{\mu}$ is determined up to a vector in the null space of $\hat{\mathbf{M}}$, generating fewer indeterminacies than the first method.

3.3 Factorization of Löwner matrices

There is well-known theory about the factorization of Hankel matrices with the Vandermonde decomposition [48, 101, 376]. Each Hankel matrix $\mathbf{H} \in \mathbb{C}^{I \times J}$ can

be written as $\mathbf{V}\mathbf{G}\tilde{\mathbf{V}}^{\mathrm{T}}$ with Vandermonde matrices $\mathbf{V} \in \mathbb{C}^{I\times H}$ and $\tilde{\mathbf{V}} \in \mathbb{C}^{J\times H}$, and a block-diagonal matrix $\mathbf{G} \in \mathbb{C}^{H\times H}$. The rank H of $\mathbf{H}$ is the degree of the underlying exponential polynomial. For Löwner matrices, a general factorization has been developed in [152, 382]. We present the factorization in a way that facilitates the use in signal processing for modeling signals by rational functions of low degree. The factorization relies on partial fractions and Cauchy matrices. A Cauchy matrix $\mathbf{C}_{\mathbf{u},\mathbf{v}} \in \mathbb{K}^{I\times J}$ based on two vectors $\mathbf{u} \in \mathbb{K}^I$, $\mathbf{v} \in \mathbb{K}^J$ with $\forall i,j : u_i \neq v_j$ consists of elements $c_{i,j} = 1/(u_i - v_j)$.

We assume a rational source $s(t)$ with the following partial fraction decomposition:

$$s(t) = a(t) + \sum_{f=1}^{F}\sum_{d=1}^{D_f} \frac{c_{f,d}}{(t-p_f)^d}, \tag{3.5}$$

meaning that $s(t)$ has F complex poles p_f which can have a multiplicity D_f higher than one. Equation (3.5) is general in the sense that it covers all rational functions. The first term $a(t)$ in (3.5) denotes a polynomial of degree W. We define $L = W + \sum_{f=1}^{F} D_f$. As a working assumption, W is zero for most sources; Section 3.4 considers the uniqueness conditions.

In each of the three following subsections we illustrate the factorization of the Löwner matrices corresponding to (3.5) in a constructive way, first discussing the case of non-coinciding poles and afterwards discussing coinciding poles, i.e., poles with a multiplicity higher than one. In a third subsection polynomials are discussed. Subsection 3.3.4 concludes the section by applying the results to blind signal separation.

3.3.1 Case of rational functions with non-coinciding poles

Consider a rational function with $W = 0$ and with F poles p_f with multiplicities $D_f = 1$ for $1 \leq f \leq F$ (thus $L = F$), collected in a vector $\mathbf{p}$ and with corresponding coefficients c_f:

$$s(t) = a + \sum_{f=1}^{F} \frac{c_f}{t-p_f}.$$

If sampled in a point set T with N distinct points and partitions X and Y, the corresponding Löwner matrix is given by:

$$L_{i,j} = \left(\sum_{f=1}^{F} \frac{c_f}{x_i - p_f} - \sum_{f=1}^{F} \frac{c_f}{y_j - p_f} \right) / (x_i - y_j),$$

$$= \sum\nolimits_{f=1}^{F} -c_f \cdot \frac{1}{x_i - p_f} \cdot \frac{1}{y_j - p_f} \qquad \forall i, j.$$

Hence, the matrix $\mathbf{L}$ can be factorized as

$$\mathbf{L} = \mathbf{Z} \cdot \mathrm{diag}(-c_1, \ldots, -c_F) \cdot \tilde{\mathbf{Z}}^{\mathrm{T}},$$

with

$$\mathbf{Z} = \begin{bmatrix} \frac{1}{x_1 - p_1} & \cdots & \frac{1}{x_1 - p_F} \\ \vdots & \ddots & \vdots \\ \frac{1}{x_I - p_1} & \cdots & \frac{1}{x_I - p_F} \end{bmatrix}, \; \tilde{\mathbf{Z}} = \begin{bmatrix} \frac{1}{y_1 - p_1} & \cdots & \frac{1}{y_1 - p_F} \\ \vdots & \ddots & \vdots \\ \frac{1}{y_J - p_1} & \cdots & \frac{1}{y_J - p_F} \end{bmatrix}.$$

One can see that $\mathbf{Z} = \mathbf{C}_{\mathbf{x},\mathbf{p}}$ and $\tilde{\mathbf{Z}} = \mathbf{C}_{\mathbf{y},\mathbf{p}}$.

The assumption of non-coinciding poles with its factorizaton of the Löwner matrix suffices in many cases. We present the general case with coinciding poles in the following subsection for the sake of completeness.

3.3.2 General case of rational functions with coinciding poles

We first study the Löwner matrix of the term corresponding to $\sum_{d=1}^{D_f} \frac{c_{f,d}}{(t-p_f)^d}$ in (3.5), i.e., for a pole p_f with multiplicity $D_f \geq 1$. The matrix is given by

$$L_{i,j} = \left(\sum_{d=1}^{D_f} \frac{c_{f,d}}{(x_i - p_f)^d} - \sum_{d=1}^{D_f} \frac{c_{f,d}}{(y_j - p_f)^d} \right) / (x_i - y_j),$$

$$= \sum_{d=1}^{D_f} c_{f,d} \frac{(y_j - p_f)^d - (x_i - p_f)^d}{(x_i - y_j)(x_i - p_f)^d (y_j - p_f)^d}, \qquad \forall i, j.$$

Because $(x_i - y_j) = ((x_i - p_f) - (y_j - p_f))$, one can derive

$$L_{i,j} = \sum_{d=1}^{D_f} \left(-c_{f,d} \sum_{e=1}^{d} \frac{1}{(x_i - p_f)^e (y_j - p_f)^{d-e+1}} \right).$$

The result yields the factorization $\mathbf{L} = \mathbf{Z}_{f,D_f}\mathbf{G}_{f,D_f}\tilde{\mathbf{Z}}^{\mathrm{T}}_{f,D_f}$ with $\mathbf{Z}_{f,D_f}$ and $\tilde{\mathbf{Z}}_{f,D_f}$ being Vandermonde matrices:

$$\mathbf{Z}_{f,D_f} = \begin{bmatrix} \frac{1}{x_1-p_f} & \cdots & \frac{1}{(x_1-p_f)^{D_f}} \\ \vdots & \ddots & \vdots \\ \frac{1}{x_I-p_f} & \cdots & \frac{1}{(x_I-p_f)^{D_f}} \end{bmatrix},$$

$$\tilde{\mathbf{Z}}_{f,D_f} = \begin{bmatrix} \frac{1}{y_1-p_f} & \cdots & \frac{1}{(y_1-p_f)^{D_f}} \\ \vdots & \ddots & \vdots \\ \frac{1}{y_J-p_f} & \cdots & \frac{1}{(y_J-p_f)^{D_f}} \end{bmatrix}.$$

These matrices are variants of the confluent Cauchy matrices from [382]. We also have

$$\mathbf{G}_{f,D_f} = \begin{bmatrix} -c_{f,1} & -c_{f,2} & -c_{f,3} & \cdots & -c_{f,D_f} \\ -c_{f,2} & -c_{f,3} & -c_{f,4} & \cdots & 0 \\ -c_{f,3} & -c_{f,4} & -c_{f,5} & \cdots & 0 \\ \vdots & \vdots & \vdots & \iddots & \vdots \\ -c_{f,D_f} & 0 & 0 & \ldots & 0 \end{bmatrix}.$$

3.3.3 Case of polynomials

Let $s(t)$ be a polynomial of degree W, i.e., $s(t) = \sum_{w=1}^{W} a_w t^w = a_W t^W + \ldots + a_1 t + a_0$, with $a_w \in \mathbb{K}$, $1 \leq w \leq W$. The Löwner matrix corresponding to a point set T with partitions X and Y is given by:

$$\begin{aligned} L_{i,j} &= \left(\sum_{w=1}^{W} a_w x_i^w - \sum_{w=1}^{W} a_w y_j^w \right) / (x_i - y_j), \\ &= \sum_{w=1}^{W} a_w \sum_{v=0}^{w-1} x_i^v y_j^{w-v-1}, \qquad \forall i,j. \end{aligned}$$

The matrix admits to a factorization $\mathbf{L} = \mathbf{Z}_W \mathbf{G}_W \tilde{\mathbf{Z}}_W^{\mathrm{T}}$ with $\mathbf{Z}_W \in \mathbb{K}^{I \times W}$, $\tilde{\mathbf{Z}}_W \in \mathbb{K}^{J \times W}$:

$$\mathbf{Z}_W = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{W-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_I & \cdots & x_I^{W-1} \end{bmatrix}$$

$$\tilde{\mathbf{Z}}_W = \begin{bmatrix} 1 & y_1 & \cdots & y_1^{W-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & y_J & \cdots & y_J^{W-1} \end{bmatrix} \tag{3.6}$$

and with

$$\mathbf{G}_W = \begin{bmatrix} a_1 & a_2 & \cdots & a_W \\ a_2 & a_3 & \cdots & 0 \\ \vdots & \vdots & \iddots & \vdots \\ a_W & 0 & \dots & 0 \end{bmatrix} \in \mathbb{K}^{W \times W}.$$

3.3.4 Summary and implication for blind source separation

Regarding the complete rational signal $s(t)$ from (3.5), its associated Löwner matrix $\mathbf{L}$ admits to a general decomposition

$$\mathbf{L} = \mathbf{Z}\mathbf{G}\tilde{\mathbf{Z}}^{\mathrm{T}} \tag{3.7}$$

in which

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_W & \mathbf{Z}_{1,D_1} & \mathbf{Z}_{2,D_2} & \cdots & \mathbf{Z}_{F,D_F} \end{bmatrix} \in \mathbb{K}^{I \times L}$$

$$\tilde{\mathbf{Z}} = \begin{bmatrix} \tilde{\mathbf{Z}}_W & \tilde{\mathbf{Z}}_{1,D_1} & \tilde{\mathbf{Z}}_{2,D_2} & \cdots & \tilde{\mathbf{Z}}_{F,D_F} \end{bmatrix} \in \mathbb{K}^{J \times L}$$

with $L = W + \sum_{f=1}^{F} D_f$. The matrix $\mathbf{G} \in \mathbb{K}^{L \times L}$ is a block-diagonal matrix with Hankel and upper antitriangular matrices $\mathbf{G}_W$ and $\mathbf{G}_{f,D_f}$ for $1 \leq f \leq F$ on its diagonal.

Instead of a single signal, suppose we have R different sources $s_r(t)$. From this point on, we use the subscript r to denote the specific source. Each Löwner matrix $\mathbf{L}_r$, constructed with the same point set partitions admits to a decomposition of the form (3.7).

Let us assume that $I, J \geq \max(L_1, \ldots, L_R)$. First, the matrices $\mathbf{Z}_r$ and $\tilde{\mathbf{Z}}_r$ clearly have full rank for distinct points. Second, $\mathbf{G}_r$ is nonsingular since its diagonal blocks are nonsingular. Each Löwner matrix $\mathbf{L}_r$ is then of rank L_r and the rth term in (3.3) is rank-$(L_r, L_r, 1)$ for $1 \leq r \leq R$. Hence, eq. (3.3) is a decomposition of $\mathcal{L}_{\mathbf{X}}$ in rank-$(L_r, L_r, 1)$ terms. Furthermore, the matrix multiplication $\mathbf{A}_r\mathbf{B}_r^{\mathrm{T}}$ in (3.4) can be written as $\mathbf{Z}_r\mathbf{G}_r\tilde{\mathbf{Z}}_r^{\mathrm{T}}$ with the previously explained structure.

The next section will explain whether (and when) this decomposition in rank-$(L_r, L_r, 1)$ terms is unique.

3.4 Uniqueness

The analysis in this section is the Löwner counterpart of the Hankel case[101]. We first recall [104, Theorem 4.1] regarding the uniqueness of a BTD in rank-$(L_r, L_r, 1)$ terms:

Theorem 3.2. *Consider a decomposition of $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ in rank-$(L_r, L_r, 1)$ terms as in* (3.1) *and* (3.2)*, with $I, J \geq \sum_{r=1}^{R} L_r$. If $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_R \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \mathbf{B}_R \end{bmatrix}$ have full column rank and $\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 & \cdots & \mathbf{c}_R \end{bmatrix}$ does not have proportional columns, then the decomposition is essentially unique.*

We call the decomposition 'essentially unique' when one can only permute the rth and r'th terms in (3.1) when $L_r = L_{r'}$ and when one can only scale $\mathbf{E}_r$ provided that $\mathbf{c}_r$ is counterscaled. In [101] the theorem is generalized into the following theorem for the block term decomposition, including a necessary condition:

Theorem 3.3. *Consider a decomposition of $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ in rank-$(L_r, L_r, 1)$ terms as in* (3.1) *and* (3.2)*. Define $\mathbf{W}(\mathbf{w}) = \sum_{r=1}^{R} w_r \mathbf{E}_r$. Assume the following conditions to be satisfied:*

(C1) For every $\mathbf{w}$ that has at least two nonzero entries, we have that $\operatorname{rank}(\mathbf{E}(\mathbf{w})) > \max_{r|w_r \neq 0}(L_r)$.

(C2) The columns of $\mathbf{C}$ are linearly independent.

The decomposition is then essentially unique. On the other hand, if condition (C1) is not satisfied, then the decomposition is not essentially unique.

We now apply Theorem 3.2 to BSS for rational sources with coinciding poles and polynomial terms:

Theorem 3.4. *Consider a matrix* $\mathbf{M} \in \mathbb{K}^{K \times R}$ *that does not have proportional columns, and a matrix* $\mathbf{S} \in \mathbb{K}^{R \times N}$ *in which every row has a structure as in* (3.5). *Assume that* $\lfloor \frac{N+1}{2} \rfloor \geq \sum_{r=1}^{R} L_r$. *If all the poles* p_{r,f_r} *are distinct for* $1 \leq f_r \leq F_r$, $1 \leq r \leq R$ *in* (3.5) *and if at most one source contains a polynomial term (at most one* $W_r \neq 0$*), then the decomposition* $\mathbf{X} = \mathbf{MS}$ *is essentially unique.*

Proof. The constraint $\lfloor \frac{N+1}{2} \rfloor \geq \sum_{r=1}^{R} L_r$ allows us to map the rows of $\mathbf{X}$ to Löwner matrices with sizes $I \times J$ and with $I, J \geq \sum_{r=1}^{R} L_r$. With distinct poles in (3.5), the matrices $\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{r=1} & \mathbf{Z}_{r=2} & \cdots & \mathbf{Z}_{r=R} \end{bmatrix} \cdot \text{diag}(\mathbf{G}_{r=1}, \ldots, \mathbf{G}_{r=R})$ and $\tilde{\mathbf{Z}} = \begin{bmatrix} \tilde{\mathbf{Z}}_{r=1} & \tilde{\mathbf{Z}}_{r=2} & \cdots & \tilde{\mathbf{Z}}_{r=R} \end{bmatrix}$ have full column rank. This is assuming that at most one $W_r \neq 0$, i.e., at most one $\mathbf{Z}_{W_r}$ and $\tilde{\mathbf{Z}}_{W_r}$ from (3.6) is included in $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$, respectively. Indeed, $\mathbf{Z}_{W_r}$ and $\mathbf{Z}_{W_{r'}}$ from (3.6) have mutually linear dependent columns for $r \neq r'$ (likewise for $\tilde{\mathbf{Z}}_{W_r}$). The uniqueness result then follows from Theorem 3.2. □

The proof shows that it is not possible to separate polynomials, as pointed out in [101] too. A polynomial from a single source can be identified though.

It is also important to remark that if the sources have distinct poles, uniqueness of the factorization $\mathbf{X} = \mathbf{MS}$ is guaranteed when enough samples are available. Even so, only $2 \times \sum_{r=1}^{R} L_r$ samples are needed, with L_r mostly small.

Furthermore, uniqueness results can be obtained if the sources share common poles too [101]. Part 3.6.4 gives an example with the second fetal electrocardiogram experiment. A sufficient property can be found in [346, condition (5.18)].

Other general and useful results are presented in [345], while the procedure from [132] can be used to deduce generic uniqueness conditions.

3.5 Connection with Hankel-based tensorization and Vandermonde decomposition

In Subsection 3.5.1 we give a connection between Löwner and Hankel matrices, with the latter being used in blind signal separation of exponential polynomials [101]. A discussion about the choice of sampling points is given in a second subsection, illustrating that an equidistant sampling is not needed for Löwner-based BSS.

3.5.1 Transformation between Löwner and Hankel

A strong connection between Löwner and Hankel matrices was given in [152] and afterwards generalized in [381] and used in [382]. We repeat the theorem in a customized way:

Theorem 3.5. *For fixed point sets X and Y with distinct points, the mapping $\mathcal{F} : \mathbf{H} \to \mathbf{L} = \mathbf{W_x} \mathbf{H} \mathbf{W_y}^T$ is an isomorphism (i.e. there is a one-to-one relationship) between the class of all $I \times J$ Hankel matrices and the class of $I \times J$ Löwner matrices corresponding to X and Y, with*

$$(\mathbf{W_x})_{k,l} = \frac{1}{l!} \frac{\mathrm{d}^l a_k(z)}{\mathrm{d}z^l}\bigg|_{z=0}, \quad a_k(z) = \prod_{i \neq k} (z - x_i)\,;$$

$$(\mathbf{W_y})_{k,l} = \frac{1}{l!} \frac{\mathrm{d}^l b_k(z)}{\mathrm{d}z^l}\bigg|_{z=0}, \quad b_k(z) = \prod_{j \neq k} (z - y_j)\,.$$

The matrices $\mathbf{W_x}$ and $\mathbf{W_y}$ depend only on the point sets X and Y and not on the actual signals. As the matrices $\mathbf{W_x}$ and $\mathbf{W_y}$ are of full column rank in the generic case, an important consequence of the isomorphism is that $\mathrm{rank}(\mathbf{L}) = \mathrm{rank}(\mathbf{H})$. The relationship enables, for example, the transposition of uniqueness results between the different techniques.

The condition number of $\mathbf{W_X}$ and $\mathbf{W_Y}$ depends heavily on the point set however. For equidistant samples on the real axis, the matrices are highly ill-conditioned, so that the explicit use of the transformation can pose numerical difficulties.

3.5.2 Choice of sampling points

The results obtained in the previous sections (such as Theorem 3.4 or the factorizations in Section 3.3) are independent of the sample points used. Any sampling in the complex domain can be used.

For the Hankel case, one needs measurements sampled in an equidistant way; otherwise the Vandermonde decomposition is not applicable. This is a fundamental difference with respect to the Löwner case for which any collection of sampling points can be used: Chebyshev nodes, logarithmic distribution, ... The arbitrary choice of sampling points enables the use of compressed sensing techniques. One can, for example, randomly sample the observed signals at a number of time instances, useful in the case of high-cost measurement settings.

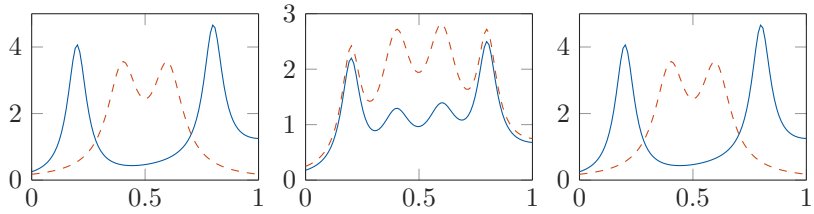


Figure 3.3: Results for the first experiment of Section 3.6.1. Left: the two original sources. Middle: the observed signals. Right: the perfectly recovered sources.

3.6 Experiments and applications

In the first Subsection 3.6.1 an example of a separation is given with one source containing a polynomial term. We also investigate the behavior for a low number of samples and with respect to the presence of noise. Subsection 3.6.2 presents the separation for an underdetermined mixture with more source signals than observed signals. A third Subsection 3.6.3 explains the illustration of fluorescence spectroscopy. We conclude with a description of the use of Löwner-based tensorization for fetal electrocardiogram extraction in Subsection 3.6.4.

To present the recovered sources and to determine the relative error, we use an optimal scaling and permutation step (the default indeterminacies in BSS) with respect to the theoretical sources. The relative error is then defined as the relative difference in Frobenius norm, e.g., if $\hat{\mathbf{S}}$ are the recovered sources after this step, we have a relative error $\epsilon_{\mathbf{S}} = \|\mathbf{S} - \hat{\mathbf{S}}\|/\|\mathbf{S}\|$. Second, the signal-to-noise ratio (SNR) is defined as the power of the signal to the power of the noise, with the noise being Gaussian additive noise (unless stated otherwise). To calculate a BTD in rank-$(L_r, L_r, 1)$ terms, various approaches similar to CPD algorithms exist such as alternating least squares, unconstrained nonlinear optimization, or nonlinear least squares [113, 332, 333]. We employ the latter by using Tensorlab [334]. A generalized eigenvalue decomposition is used for the initialization [104]. In all experiments only a few iterations are needed to reach convergence. For information about complexity we refer to [332, 333]. By default, the sampling points are chosen equidistantly on the real axis in $[0, 1]$. To construct the Löwner matrix, we use square matrices and partition the point set T in two interleaved partitions X and Y. An extensive analysis did not give a clear answer on which partitioning method is preferred; both methods described in section 3.2.2 give a similar performance. The sources are by default recovered by using the inverted mixing matrix, except for the underdetermined case in subsection 3.6.2.

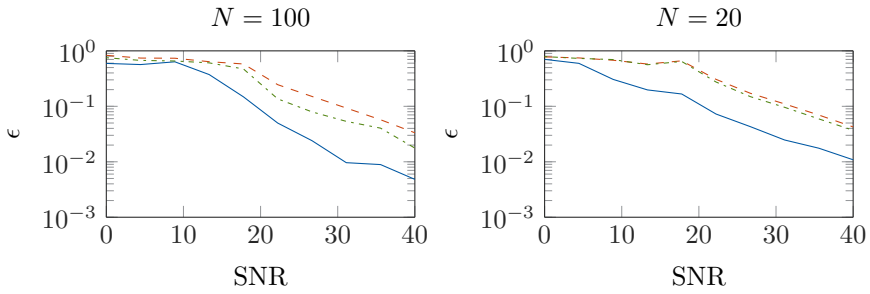


Figure 3.4: Recovery results for 100 and 20 samples when Gaussian noise is added, in function of SNR. The median relative errors across 100 experiments are shown, for the mixing matrix (——), and for the recovered sources for the two different recovery methods of Section 3.2.4: by using the inverted mixing matrix (- - -) and by using the Löwner matrices (·····).

3.6.1 General experiment

We start with the separation of the two following sources:

$$s_1(t) = \left((t-0.2)^2 + 0.05^2\right)^{-1} + \left((t-0.8)^2 + 0.05^2\right)^{-1} + t^2,$$

$$s_2(t) = \left((t-0.4)^2 + 0.08^2\right)^{-1} + \left((t-0.6)^2 + 0.08^2\right)^{-1}.$$

The first source of degree 6 has two conjugated pole pairs $0.2 \pm 0.05j$ and $0.8 \pm 0.05j$, and also includes a polynomial term of degree 2. The second source has two conjugated pole pairs $0.4 \pm 0.08j$ and $0.6 \pm 0.08j$ and has degree 4. The two sources are divided by a factor 100 and 50, respectively, to obtain suitable magnitudes. Fig. 3.3 shows the signals. A mixture with $\mathbf{M} = [0.5,\ 0.3;\ 0.5,\ 0.7]$ is used, and $L_1 = 6$ and $L_2 = 4$. In a first case, we take 100 equidistant samples in $[0, 1]$. The recovered sources for the noiseless case are presented in Fig. 3.3 and the results for the noisy case are shown in Fig. 3.4 on the left. A second case, presented in function of SNR in Fig. 3.4 on the right, only uses 20 samples.

Fig. 3.5 shows the results for an experiment in which the number of source signals is varied, given ten observed signals. The ith source is given by $s_i(t) = \left((t-r_i)^2 + q^2\right)^{-1}$, with r_i equidistantly spaced in $[0, 1]$. Two cases are considered: $q = 0.01$ (mildly overlapping) and $q = 0.1$ (highly overlapping). An SNR of 25 is used. $\mathbf{M}$ is taken column-wise orthonormal.

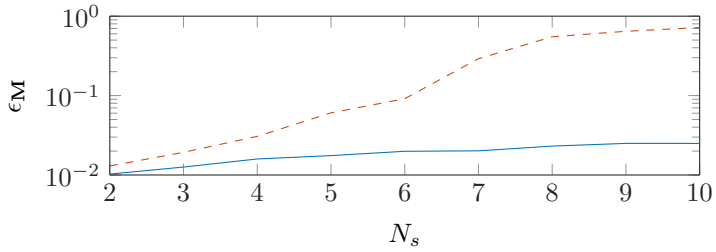


Figure 3.5: Recovery results for a varying number of mildly (——) and highly (- - -) overlapping source signals. Median relative errors across 100 experiments are shown.

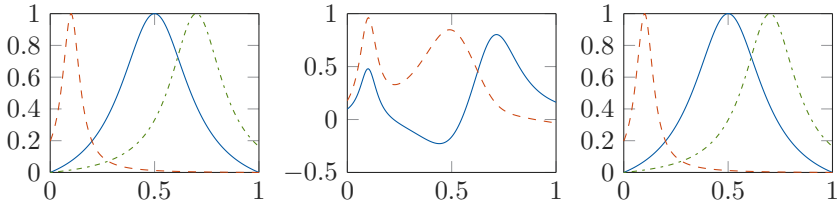


Figure 3.6: Results for the underdetermined experiment. Left: the three original sources. Middle: the two observed mixed signals. Right: the three recovered sources after optimal scaling and permutation. A perfect recovery has been obtained of more sources than observed signals.

3.6.2 Underdetermined system

We examine the identification of more sources than there are observed signals available. Consider the following three source signals:

$$s_1(t) = \left((t-0.1)^2 + 0.05^2\right)^{-1} \quad \text{(pole pair } 0.1 \pm 0.05j\text{)}$$

$$s_2(t) = \left((t-0.5)^2 + 0.20^2\right)^{-1} \quad \text{(pole pair } 0.5 \pm 0.20j\text{)}$$

$$s_3(t) = \left((t-0.7)^2 + 0.15^2\right)^{-1} \quad \text{(pole pair } 0.7 \pm 0.15j\text{)}$$

sampled in $t \in [0, 1]$ with $N = 100$ equidistant points. The signals are mixed into two observed signals using the mixing matrix $\mathbf{M} = [-0.5,\ 0.5,\ 1;\ 0.9,\ 0.9,\ -0.2]$. We use $L_1 = L_2 = L_3 = 2$. A perfect reconstruction of the three sources is obtained, as Fig. 3.6 illustrates.

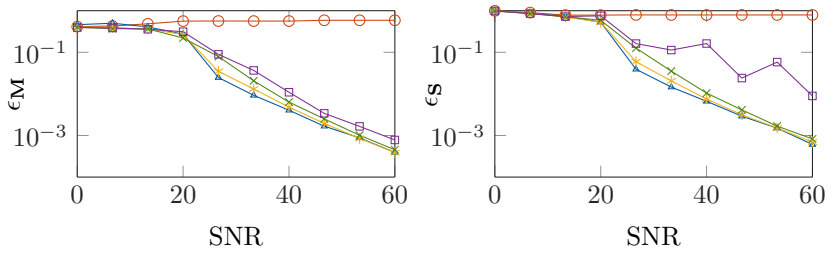


Figure 3.7: Results for the underdetermined mixture, in function of SNR and for varying ranks with (L_1, L_2, L_3) being (2,1,1) (—○—); (2,2,2) (—△—); (3,2,2) (—✳—); (4,3,2) (—×—) and (4,4,4) (—□—). The sources are determined up to a constant, cf. subsection 3.2.4. Median relative errors across 100 experiments are shown.

In Fig. 3.7 we include results for varying SNR and different values of (L_1, L_2, L_3). Note that several choices of the degrees lead to good results. This shows that the choice of the (multilinear) rank(s) is not very critical [101, 102]. A trial-and-error method can be used to deduct L_r, knowing that the multilinear rank of $\mathcal{L}_{\mathbf{X}}$ is bounded by $(\sum_{r=1}^{R} L_r, \sum_{r=1}^{R} L_r, R)$.

3.6.3 Fluorescence spectroscopy

Source separation, also known as curve resolution, is a valuable technique used in fluorescence spectroscopy. It enables the estimation of relative concentrations and pure analyte spectra from fluorescence measurements of chemical analytes in mixtures. Consider a sufficiently diluted chemical solution containing different amounts of R chemical components. By exciting the mixture at K different excitation wavelengths and measuring the spectrum of the emitted light at N different emission wavelenghts, one obtains an intensity matrix called $\mathbf{X} \in \mathbb{R}^{K \times N}$. Through the Beer-Lambert law [241], the spectra of the mixture are linearly dependent on the spectra of the underlying chemical components and on their concentrations, and one can show that $\mathbf{X} = \mathbf{A}\mathbf{\Sigma}\mathbf{B} = \mathbf{M}\mathbf{S}$ with $\mathbf{A} \in \mathbb{R}^{K \times R}$ containing the excitation spectra of the R underlying chemical components in the columns, $\mathbf{B} \in \mathbb{R}^{R \times N}$ containing the emission spectra of the components in the rows, and $\mathbf{\Sigma} \in \mathbb{R}^{R \times R}$ a diagonal matrix with the concentrations. When interpreted in terms of source signals in $\mathbf{S}$ with a mixture matrix $\mathbf{M}$, the matrix $\mathbf{\Sigma}$ can be included both in $\mathbf{S}$ or $\mathbf{M}$. We allocate the emission spectra (rather than the excitation spectra) to the source signals. As the different spectra of the underlying components are unknown, this is a standard BSS problem.

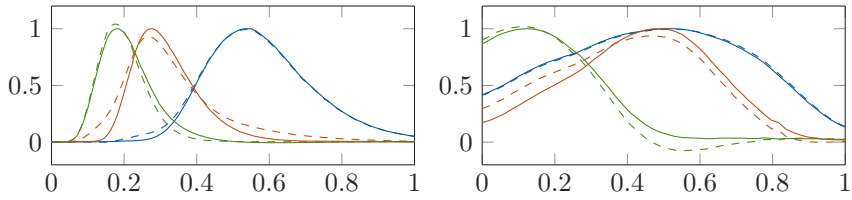


Figure 3.8: Results for the fluorescence experiment. On the left the emission spectra are shown (being approximated with rational functions of degree 2) and on the right the excitation spectra are shown. The real signals are given by solid lines (——) and the reconstructed signals by dashed lines (- - -).

Typical techniques using multilinear algebra resort to the use of multiple mixtures: the different excitation-emission matrices (EEM) are stacked, and a solution is obtained with a CPD [19, 57, 328]. Our technique only requires a single EEM, reducing the measurement cost and enabling analysis when only a single EEM is available. This is done with the understanding that the emission spectra can be well approximated by rational functions. It is also a possible alternative to time-dependent spectroscopy techniques.

The dataset[1] used in this chapter contains the components phenylalanine, tyrosine and tryptophan [56, 57, 216]. The measurements are done with excitation wavelengths in 260-300 nm and emission wavelengths in 250-450 nm, both with steps of 1 nm. We mix the analytes with concentrations of 0.5, 0.2 and 0.3, respectively. In Fig. 3.8 the pure emission and excitation spectra of the components are visualized; the emission spectra have been Löwnerized and approximated by rational functions of a significantly low degree. The technique is used to separate the observations in three components with $L_r = 2$ for $r = 1, 2, 3$. The results are given in Fig. 3.8, and one can see that the components are separated very well given that only a single mixture has been used. A relative error on the emission (excitation) spectra of 0.1094 (0.096) has been obtained. For comparison, ICA yields a relative error of 0.640 (0.305), 0.818 (0.491) and 0.606 (0.682) for FastICA, RobustICA and JADE, respectively.

A final remarkable thing to mention is that in theory, we do not need fluorescence measurements across 50 different excitation wavelengths. The more observations in practice however, the more accurate the separation will be. Fig. 3.9 illustrates the findings if less than 50 excitation wavelengths are used. The extracted rows from the observed data matrix are selected as equidistant as possible.

[1] Available from `http://www.models.kvl.dk/Amino_Acid_fluo`

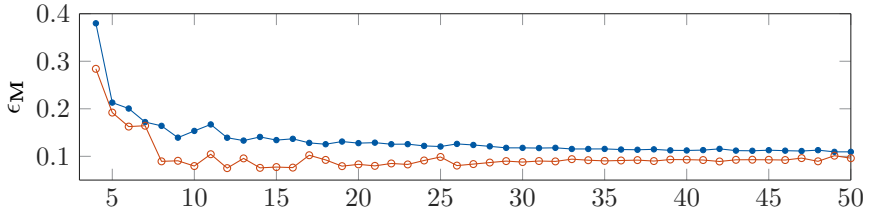


Figure 3.9: Results for the fluorescence experiment of the relative error on the mixing matrix (—•—) and the recovered sources (—○—) in function of the number of excitations used.

3.6.4 Fetal electrocardiogram extraction

In this application, the proposed technique is used for the extraction of antepartum fetal electrocardiogram (fetal ECG or FECG) from multilead cutaneous potential recordings. While examining ECG recordings measured on the pregnant woman's skin (cutaneous), one tries to eliminate the dominant heartbeat of the mother. Seeing the problem as a blind signal separation problem, one can resort to the use of ICA [111]. ICA falls short however when only few samples or heartbeats are available. Second, for coinciding beats, the basic independence assumption of ICA is not valid.

It seems that ECG beats (with their easily recognizable QRS complexes) can be well modeled by rational functions [154, 257, 258]. The Löwner technique needs no preprocessing, as opposed to the technique in, e.g., [212]. We carry out two experiments with real-life datasets to illustrate the technique.

The first dataset consists of 8 measurement signals (of which 5 abdominal and 3 thoracic signals), available at DaISy[2] [65, 111]. For the sake of simplicity, only the 5 abdominal signals and only the first 500 samples are used, with the observations shown in Fig. 3.10. Each signal has been scaled to unit norm. For recovery, a separation into two source signals is not enough and at least three source signals are needed; this is also the case when applying ICA [111]. For the BTD, a rank of 20 for each source signal has been used. The three recovered sources are visualized in Fig. 3.11 with a clear separation of the two different ECG sources.

The second dataset contains a limited number of heartbeats with the beats of the mother and fetus coinciding. A mixing matrix $\mathbf{M} = [1\ 1; 1\ -0.8]$ is used to mix the signals. Fig. 3.12 visualizes the signals and the recovered sources. When using the proposed technique with again a rank 20 for each source signal, an

[2]Available from `http://homes.esat.kuleuven.be/~smc/daisy/daisydata.html`.

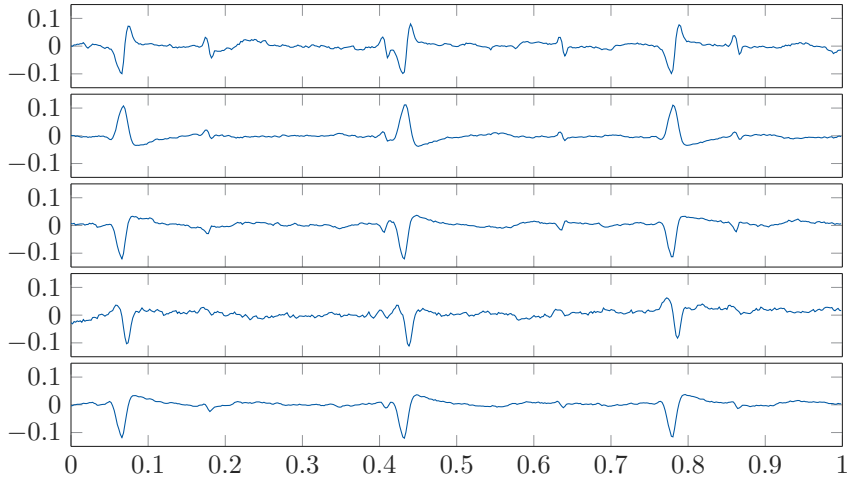


Figure 3.10: Visualization of the 5 abdominal ECG recordings used in the first FECG experiment.

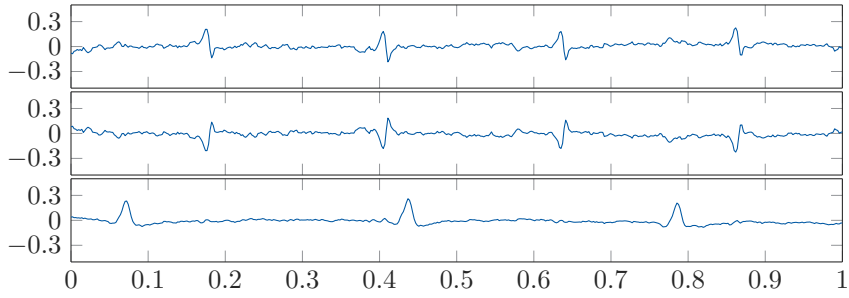


Figure 3.11: The separation of the ECG recordings into three recovered source signals for the first FECG experiment. One clearly notices the separation of the fetal heart beats (above) and the heart beats of the mother (below). Typically, the fetal heart rate is significantly higher than the mother's heart rate.

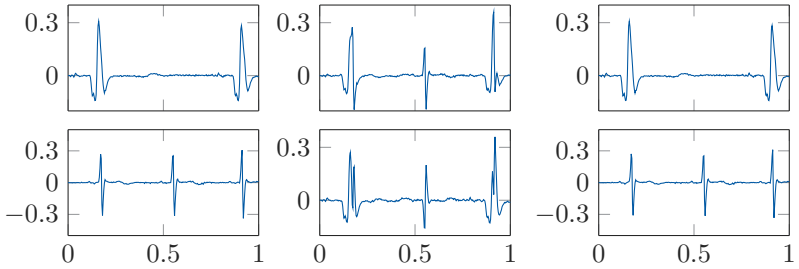


Figure 3.12: Illustrations for the second FECG experiment. Left we have a limited amount of heart beats of mother (above) and fetus (below) where some beats coincide. The mixed signals are shown in the middle. To the right, the recovered sources are shown. An excellent recovery is obtained with a relative error on the sources of only 0.013.

excellent recovery is obtained with a relative source error of 0.013. To compare, ICA recovers the signals up to a relative error of 0.126, 0.125 and 0.29 for FastICA, RobustICA and JADE, respectively.

3.7 Conclusion

A novel technique for blind signal separation has been proposed for signals that can be modeled as rational functions. The tensor-based technique makes use of a deterministic tensorization with Löwner matrices and the obtained tensor is decomposed with a block term decomposition. In this chapter, the factorization of the Löwner matrices has been analyzed, together with the uniqueness conditions. The proposed method can be applied to any collection of sampling points and not only for equidistant points, as has been discussed while relating the method to another separation technique using a source model of exponential polynomials. Two synthetic experiments (including an underdetermined mixture) and two real-life illustrations with fluorescence spectroscopy and fetal electrocardiogram extraction have been used to verify the proposed technique. The method has been compared against ICA, demonstrating the power of the deterministic Löwner technique when the source signals are not independent or when only a limited number of samples are available.

Chapter 4

A tensor-based method for large-scale blind source separation using segmentation

Abstract Many real-life signals are compressible, meaning that they depend on much fewer parameters than their sample size. In this chapter we use low-rank matrix or tensor representations for signal compression. We propose a new deterministic method for blind source separation that exploits the low-rank structure, enabling a unique separation of the source signals and providing a way to cope with large-scale data. We explain that our method reformulates the blind source separation problem as the computation of a tensor decomposition, after reshaping the observed data matrix into a tensor. This deterministic tensorization technique is called segmentation and is closely related to Hankel-based tensorization. We apply the same strategy to the mixing coefficients of the blind source separation problem, as in many large-scale applications the mixture is also compressible because of many closely located sensors. Moreover, we combine both strategies, resulting in a general technique that allows us to exploit the underlying compactness of the sources and the mixture simultaneously. We illustrate the techniques for fetal electrocardiogram extraction and direction-of-arrival estimation in large-scale antenna arrays.

Reference This chapter is a slightly adapted version of the article [51]. Changes are limited to layout and representation aspects. The candidate provided extensive guidance to the first author regarding the research and the article preparation.

4.1 Introduction

In blind source separation (BSS) one tries to reconstruct a set of unobserved sources based only on a set of observed signals. In this chapter, the latter are unknown linear instantaneous mixtures of the sources. Applications can be found in telecommunications, signal processing and biomedical sciences [83, 89, 200, 203]. In general, there is no unique solution to the BSS problem, hence, one imposes additional assumptions.

A well-known BSS method, called independent component analysis (ICA), assumes statistically independent sources [89]. Several ICA methods use higher-order statistics (HOS) in order to tensorize the BSS problem and then apply a tensor decomposition to uniquely identify the sources. Recently, a class of deterministic methods has been proposed that do not use (higher-order) statistics but assume that the sources can be modeled as, e.g., exponential polynomials or rational functions [101, 124]. Specific tensorization techniques can be used, such as Hankel-based or Löwner-based tensorization [118]. The source signals can then be uniquely recovered by block component analysis (BCA). BCA is a framework based on block term decompositions which was introduced in [102–104]. These methods, as well as the method we propose here, go further than dictionary-based methods. In the latter, one defines *a priori* a fixed signal dictionary in which one assumes the sources can be described sparsely and then one exploits this sparse representability to identify the sources [246, 415]. Here, we do not need an initial dictionary.

In this chapter, we introduce a new method for BSS that exploits the fact that many real-life signals are compressible, i.e., the fact that they can be described in terms of much fewer parameters than the actual number of samples [63, 67]. One way of representing signals in a (possibly very) compact way is a (higher-order) low-rank approximation of a tensorized version of the signal [171]. This can be interpreted as approximating the original signals by sums of Kronecker products of smaller vectors. This strategy is similar to tensor-based scientific computing in high dimensions [171, 286, 391], which has allowed one to solve problems in a number of unknowns that exceeds the number of atoms in the universe. It is used in a novel way for BSS in this chapter and is a key idea to handle large-scale BSS problems, i.e., problems with many sensors and/or samples. In particular, we use a deterministic tensorization technique, called segmentation, that reshapes each observed signal into a matrix (tensor) and stacks them into a (higher-order) tensor. The latter can be interpreted as a compact version of the Hankel-based tensorization mentioned above. We show that the BSS problem boils down to the computation of a decomposition of the tensor obtained by segmentation if the sources exhibit the hypothesized low-rank

structure. This yields a unique solution to the BSS problem and provides a way to cope with large-scale problems where conventional methods fall short. Also, it is illustrated that our method allows the separation of underdetermined mixtures, i.e., the separation of more sources than observed signals.

We can apply the same strategy to the mixing coefficients of the BSS problem (instead of the source signals) following a similar argument. Indeed, in the context of big data, we see a large increase in the number of sensors and/or sensor density in fields such as biomedical sciences and sensor array processing [43, 236]. The mixing coefficients are in that case often smoothly varying because of the many closely located sensors, allowing a (higher-order) low-rank approximation of a tensorized version of the mixing vectors. Conventional methods such as ICA fall short in a large-scale setting because of the exponential dependence on the order of the statistics. Exploiting low-rank structure on the mixing level was briefly discussed in [50]. In this chapter, we go further: we apply the strategy on the sources, as described above, but also apply it on both the sources and the mixture simultaneously. The latter is a natural extension that results into a more general method that exploits the hypothesized low-rank structure of the simultaneously tensorized source and mixing level of the BSS problem, enabling a unique solution for large-scale BSS.

We illustrate the proposed methods with two applications. First, we have the separation of the fetal and maternal electrocardiogram (ECG) from multilead cutaneous potential recordings. Our method allows a clear separation of the two sources. Second, we have direction-of-arrival (DOA) estimation for large uniform linear arrays in both a line-of-sight and multipath setting. Our methods provides accurate estimates, even for close DOAs. In very large-scale applications, the arrays, however, are typically non-uniform but this is outside the scope of this chapter; here, we focus on the main principles.

In the remainder of this section we introduce the notation and basic definitions. In Sections 4.2 and 4.3, we introduce a new BSS method that exploits the hypothesized compressibility of the sources and mixing vectors, respectively. We combine both strategies in Section 4.4. Simulations and applications are presented in Section 4.5. Finally, we conclude in Section 4.6.

4.1.1 Notation and definitions

Tensors, denoted by calligraphic letters (e.g., $\mathcal{A}$), are higher-order generalizations of vectors and matrices, denoted by bold lowercase (e.g., $\mathbf{a}$) and bold uppercase (e.g., $\mathbf{A}$) letters, respectively. The $(i_1, i_2, \ldots, i_N)$th entry of an Nth-order tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$, with $\mathbb{K}$ meaning $\mathbb{R}$ or $\mathbb{C}$, is denoted by $a_{i_1 i_2 \ldots i_N}$. The nth

element in a sequence is indicated by a superscript between parentheses (e.g., $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$). The unit vector with a one in the ith row is denoted as $\mathbf{e}_i$.

A mode-n vector of a tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined by fixing every index except the nth, e.g., $\mathbf{a}_{i_1 \cdots i_{n-1} : i_{n+1} \cdots i_N}$, and is a natural extension of the rows and columns of a matrix. The mode-n unfolding of $\mathcal{A}$ is a matrix $\mathbf{A}_{(n)}$ with the mode-n vectors as its columns (following the ordering convention in [223]). The vectorization of $\mathcal{A}$, denoted as $\text{vec}(\mathcal{A})$, maps each element $a_{i_1 i_2 \cdots i_N}$ onto $\text{vec}(\mathcal{A})_j$ with $j = 1 + \sum_{k=1}^{N} (i_k - 1) J_k$ and $J_k = \prod_{m=1}^{k-1} I_m$. The outer and Kronecker product are denoted by $\otimes$ and $\otimes$, respectively, and are related through a vectorization: $\text{vec}\,(\mathbf{a} \otimes \mathbf{b}) = \mathbf{b} \otimes \mathbf{a}$. A frontal slice of a third-order tensor $\mathcal{X} \in \mathbb{K}^{I \times J \times K}$, denoted by $\mathbf{X}_k$, is obtained by fixing the last index.

4.1.2 Tensor decompositions

An Nth-order tensor has rank one if it can be written as the outer product of N nonzero vectors. The rank of a tensor is defined as the minimal number of rank-1 terms that generate the tensor as their sum. The multilinear rank of an Nth-order tensor is equal to the tuple of mode-n ranks, which are defined as the ranks of the mode-n unfoldings of the tensor.

Definition 4.1. A *polyadic decomposition* (PD) writes an Nth-order tensor $\mathcal{A} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ as a sum of R rank-1 terms:

$$\mathcal{A} = \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \otimes \mathbf{u}_r^{(2)} \otimes \cdots \otimes \mathbf{u}_r^{(N)}. \tag{4.1}$$

The columns of the factor matrices $\mathbf{U}^{(n)} \in \mathbb{K}^{I_n \times R}$ are equal to the factor vectors $\mathbf{u}_r^{(n)}$ for $r = 1, \ldots, R$. The PD is called *canonical* (CPD) when R is equal to the rank of $\mathcal{A}$.

The CPD is a powerful model for several applications within signal processing, biomedical sciences, computer vision, data mining and machine learning [84, 223, 324]. The decomposition is *essentially unique* if it is unique up to trivial permutation of the rank-1 terms and scaling and counterscaling of the factors in the same rank-1 term. In general, no unique solution exists in the matrix case without additional assumptions for $R > 1$. In the higher-order case, we typically expect uniqueness under rather mild conditions. Consider a third-order tensor of rank R and size $I \times J \times K$ with factor matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$. Kruskal's condition states that the CPD is unique if [230]:

$$2R + 2 \leq k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}}. \tag{4.2}$$

The k-rank of a matrix $\mathbf{A}$ equals the largest number $k_{\mathbf{A}}$ such that any $k_{\mathbf{A}}$ columns of $\mathbf{A}$ are linearly independent. Condition (4.2) is deterministic in the sense that uniqueness is guaranteed for a particular choice of factor matrices satisfying the condition. Generic uniqueness conditions consider uniqueness with probability one when the entries of the factor matrices are drawn from absolutely continuous probability density functions. For example, condition (4.2) implies generic uniqueness if $2R + 2 \leq \min(I, R) + \min(J, R) + \min(K, R)$ as the k-rank of a generic matrix equals its smallest dimension. In general, milder conditions than Kruskal's can be obtained. Let us for instance consider the case where at least one of the tensor dimensions is not strictly smaller than R. For example, the CPD is generically unique for $\mathbb{K} = \mathbb{C}$ if [78, 131]:

$$R \leq (I-1)(J-1), \quad 3 \leq I \leq J, \quad \text{and} \quad R \leq K. \tag{4.3}$$

More generally, the CPD is generically unique (with a few known exceptions) if [79]:

$$R \leq \left\lceil \frac{IJK}{I+J+K-2} \right\rceil - 1 \quad \text{and} \quad IJK \leq 15000, \tag{4.4}$$

with $\lceil x \rceil$ the smallest integer not less than x. The bound on the number of entries IJK has only been verified numerically up to 15000 but is assumed to hold for larger number of entries as well. Condition (4.4) is equivalent with (4.3) for $R \leq K$ and $3 \leq I \leq J$.

Note that condition (4.4) involves the ratio between the number of entries in the tensor and the number of parameters in a rank-1 term (compensated for scaling). The condition states that the decomposition is unique with probability one if the number of entries is (strictly) larger than the number of parameters, i.e., if the tensor is (minimally) compressible. Our working assumption to solve the large-scale BSS problem is based on this compressibility, as will be explained further. We expect even milder uniqueness conditions when $N > 3$ [322, 345]. An overview and state-of-the-art deterministic and generic uniqueness conditions for higher-order tensors are given in [79, 130, 131, 133–136, 249, 345] and references therein. For a short introduction to CPD uniqueness we refer to [324, Section IV].

Definition 4.2. A *block term decomposition* (BTD) of a third-order tensor $\mathcal{X} \in \mathbb{K}^{I \times J \times K}$ *in multilinear rank-*$(L_r, L_r, 1)$ *terms* for $r = 1, \ldots, R$ is a decomposition of the form:

$$\mathcal{X} = \sum_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^{\mathrm{T}}) \otimes \mathbf{c}_r, \tag{4.5}$$

in which $\mathbf{A}_r \in \mathbb{K}^{I \times L_r}$ and $\mathbf{B}_r \in \mathbb{K}^{J \times L_r}$ have full column rank L_r and $\mathbf{c}_r$ is nonzero.

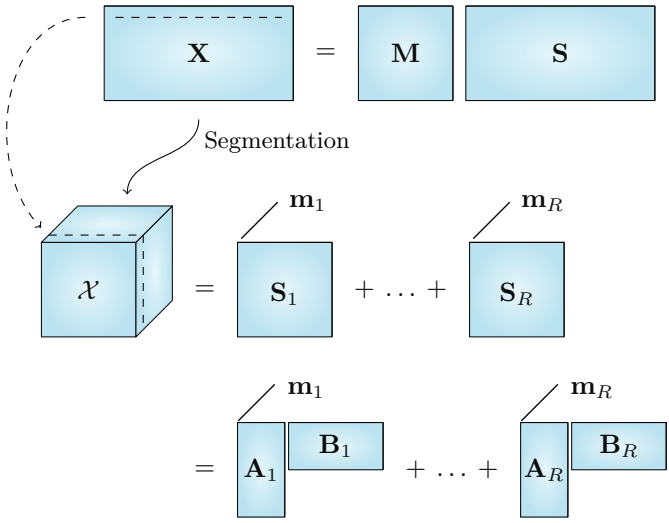


Figure 4.1: Illustration of segmentation: each row of the observed data matrix $\mathbf{X}$ is reshaped into a matrix and then stacked into a tensor $\mathcal{X}$. The reshaped sources appear in the first and second mode, and the mixing vectors appear in the third mode. The BSS problem boils down to a BTD in multilinear rank-$(L_r, L_r, 1)$ terms if the reshaped sources allow a low-rank representation, enabling a unique separation of the sources and identification of the mixing vectors.

These block terms are more general than the simple rank-1 terms of a third-order PD. Hence, they allow the modeling of more complex phenomena, see e.g., [102, 106]. Other BTDs and associated uniqueness results can be found in [101, 103, 104].

4.2 Large-scale blind source separation via low-rank sources

We derive a new BSS method that exploits the hypothesized compressibility of the sources. We show that this is possible by applying a particular deterministic tensorization technique to the observed data matrix. Decomposition of the resulting tensor allows us to uniquely retrieve the mixing vectors and the sources. In Subsections 4.2.1, 4.2.2, and 4.2.3, we define BSS, motivate the working hypothesis, and derive our method.

4.2.1 Blind source separation

We use a linear and instantaneous data model for BSS [89]:

$$\mathbf{X} = \mathbf{M}\mathbf{S} + \mathbf{N}, \tag{4.6}$$

with $\mathbf{X} \in \mathbb{K}^{M \times K}$ and $\mathbf{S} \in \mathbb{K}^{R \times K}$ containing K samples of each of the M observed and R source signals, respectively; $\mathbf{M} \in \mathbb{K}^{M \times R}$ is the mixing matrix and $\mathbf{N} \in \mathbb{K}^{M \times K}$ is the additive noise. The goal of BSS is to retrieve the unknown mixing vectors in $\mathbf{M}$ and/or the unknown sources in $\mathbf{S}$, given only the observed data $\mathbf{X}$. In the derivation of our method we ignore the noise $\mathbf{N}$ for notational simplicity, its influence will be further investigated in Section 4.5 by means of simulations.

The proposed method reshapes each observed signal, i.e., each row of $\mathbf{X}$, into a matrix and stacks them into a third-order tensor. This is illustrated in Figure 4.1. If the matricized sources admit a low-rank representation, the BSS problem can be solved uniquely by decomposing the tensorized observed data. In general, we reshape each row into an Nth-order tensor and stack them into an $(N+1)$th-order tensor. As such, the parsimonious low-rank models enable very large signal compressions, allowing one to tackle large-scale problems. In general, no unique solution to (4.6) exists without additional assumptions. By assuming that the source signals are low-rank signals, which can be written as sums of Kronecker products of smaller vectors, the problem can be reformulated as a tensor decomposition. As a decomposition of a higher-order tensor is unique under mild conditions as discussed in Subsection 4.1.2, the working assumption enables a unique solution of (4.6) under the same conditions.

4.2.2 Low-rank sources

Many real-life signals are compressible. For example, many common types of signals can be expressed in a basis such that the coefficients decay according to a power law [146]. In a large-scale setting, the amount of information contained in the signal can often be represented by a number of parameters that is much smaller than the total number of entries because there is some structure in the data [362]. Such compressible signals can often be represented in a very compact way by a low-rank approximation of a tensor representation [171, 215]; we call them *low-rank signals*. It is this notion that is the key to our approach: it enables a unique separation of the sources and identification of the mixing vectors. Moreover, it provides a way to cope with large-scale BSS problems because of the large reduction in the number of parameters. We show that our working hypothesis holds exactly for exponential polynomials.

Consider $f(t) = az^t$ evaluated in $t = 0, 1, \ldots, 5$. The resulting vector is reshaped into a (3×2) matrix $\mathbf{S}$ of rank 1:

$$\mathbf{S} = a \begin{pmatrix} 1 & z^3 \\ z & z^4 \\ z^2 & z^5 \end{pmatrix} = a \begin{pmatrix} 1 \\ z \\ z^2 \end{pmatrix} \begin{pmatrix} 1 & z^3 \end{pmatrix}. \tag{4.7}$$

The (3×4) Hankelized version $\mathbf{H}$ of the same vector is [101]:

$$\mathbf{H} = a \begin{pmatrix} 1 & z & z^2 & z^3 \\ z & z^2 & z^3 & z^4 \\ z^2 & z^3 & z^4 & z^5 \end{pmatrix} = \begin{pmatrix} 1 \\ z \\ z^2 \end{pmatrix} \begin{pmatrix} 1 & z & z^2 & z^3 \end{pmatrix}. \tag{4.8}$$

It is well-known that if the original signal is exponential, then $\mathbf{H}$ has rank one, as illustrated. One can see that the columns of $\mathbf{S}$ are a subset of the columns of $\mathbf{H}$. Hence, if $\mathbf{H}$ has rank one, then clearly $\mathbf{S}$ also has rank one. Consider now a vector $\mathbf{f} \in \mathbb{K}^K$ defined by the underlying function $f(t)$ as $f_k = f(t_k)$, $1 \leq k \leq K$, using equidistant samples. We reshape $\mathbf{f}$ into a $(I \times J)$ matrix $\mathbf{S}$ such that $\text{vec}(\mathbf{S}) = \mathbf{f}$ with $K = IJ$. Consider also a Hankelized version $\mathbf{H} \in \mathbb{K}^{I \times J_h}$ such that $h_{ij_h} = f_{i+j_h-1}$ with $K = I + J_h - 1$. Hence, we have that $\mathbf{S} = \mathbf{HQ}$ with $\mathbf{Q} \in \mathbb{K}^{J_h \times J}$ the selection matrix defined by $\mathbf{q}_j = \mathbf{e}_{(j-1)I+1}$ for $j = 1, \ldots, J$. One can verify that the matrix $\mathbf{Q}$ selects all distinct columns of $\mathbf{H}$, by comparing, e.g., the matrices in (4.7) and (4.8). It is clear that if $\mathbf{H}$ has low rank then $\mathbf{S}$ has low rank as well, while $\mathbf{S}$ offers a more compact representation than $\mathbf{H}$. It is known that $\mathbf{H}$ has low rank if the underlying functions are sums of a limited number of exponential and trigonometric terms. This fact extends to the larger class of exponential polynomials [101]. The latter allows one to model a wide range of signals in many applications, e.g., the autonomous behavior of linear systems can be described by (complex) exponential and, if we admit coinciding poles, exponential polynomials. In Table 4.1 we show the coinciding (exact) rank values of $\mathbf{H}$ and $\mathbf{S}$ for several common (exponential) polynomials; by combining such functions one can model a wide variety of signals. For example, a sine is a linear combination of two (complex conjugated) exponentials and, hence, admits a rank-2 model. Note that, while exponential polynomials can be represented by low-rank matrices, the latter allow the representation of a much larger family of signals than only exponential signals. Moreover, Hankel matrices are often ill-conditioned [368], so that the numerical rank can be significantly smaller than the theoretical one.

So far we have discussed signals that admit an exact low-rank representation. However, our approach also works well for more general compressible signals. A reshaped version of the latter often admits an approximate low-rank model as illustrated in Figure 4.2. Assume we approximate $\mathbf{S}$ by a rank-R matrix $\tilde{\mathbf{S}} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r$, then the approximation error on the original function

Table 4.1: Rank $r(\mathbf{H})$ of the Hankelized version of several (exponential) polynomials $f(t)$. If $\mathbf{H}$ has low rank then the $(I \times J)$ reshaped version $\mathbf{S}$ has low rank as well (if $R < \min(I, J)$). The latter, however, provides a much more compact representation for $f(t)$ than the former. ($p_r(t)$ is a polynomial of degree Q_r.)

$f(t)$	$r(\mathbf{H})$	$f(t)$	$r(\mathbf{H})$
az^t	1	$\sum_{r=1}^{R} a_r z_r^t$	R
$a\sin(bt)$ $a\cos(bt)$	2	$\sum_{r=1}^{R} a_r \sin(b_r t)$	$2R$
$az^t \sin(bt)$	2	$\sum_{r=1}^{R} a_r z_r^t \sin(b_r t)$	$2R$
$p(t) = \sum_{q=0}^{Q} a_q t^q$	$Q+1$	$\sum_{r=1}^{R} p_r(t)$	$\sum_{r=1}^{R} Q_r + R$
$p(t)z^t$	$Q+1$	$\sum_{r=1}^{R} p_r(t) z_r^t$	$\sum_{r=1}^{R} Q_r + R$
$p(t)\sin(at)$	$2Q+2$	$\sum_{r=1}^{R} p_r(t)\sin(a_r t)$	$\sum_{r=1}^{R} Q_r + 2R$
$p(t)z^t \sin(at)$	$2Q+2$	$\sum_{r=1}^{R} p_r(t) z_r^t \sin(a_r t)$	$\sum_{r=1}^{R} Q_r + 2R$

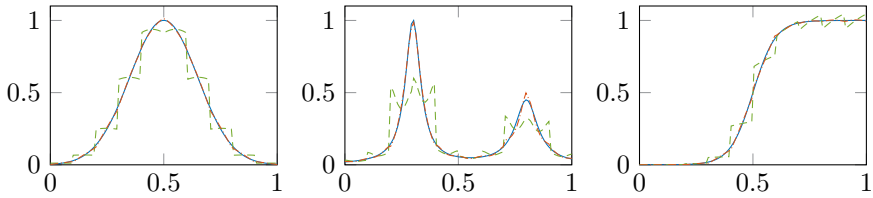


Figure 4.2: Low-rank approximation of a reshaped smooth function often provides a good representation. This is illustrated for a Gaussian (left), a rational function (middle), and a sigmoid (right) sampled uniformly 100 times in $[0, 1]$ (——). The original functions are reshaped into a (10×10) matrix and then approximated by a low-rank matrix by truncating the SVD. The reconstructed functions are obtained by vectorizing this low-rank matrix. One can clearly see that the functions can be better approximated by a rank-2 (-·-·-) than a rank-1 (- - - -) approximation.

$\mathbf{f} = \text{vec}(\mathbf{S})$ is:

$$\|\mathbf{f} - \text{vec}(\tilde{\mathbf{S}})\|_{\text{F}}^2 = \|\mathbf{f} - \sum_{r=1}^{R} \mathbf{b}_r \otimes \mathbf{a}_r\|_{\text{F}}^2. \tag{4.9}$$

Recall from Subsection 4.1.1 that a Kronecker product equals a vectorized outer product. We can make the approximation error (4.9) as small as desired by increasing R. Since (4.9) is just a vectorized version of $\|\mathbf{S}-\tilde{\mathbf{S}}\|_{\text{F}}^2$, Eckart–Young's theorem provides an upper bound on the approximation error [142]. Namely, the least squares error on the representation of the signal $\mathbf{f}$ is the sum of the squares of the discarded singular values of $\mathbf{S}$. The singular value spectrum of $\mathbf{S}$ is often fast decaying, and hence the signal $\mathbf{f}$ often admits a good representation of the form (4.9) for low R. It is outside the scope of this chapter to investigate in general under which conditions on the signal $\mathbf{f}$ the error in (4.9) is small. However, we do provide explicit bounds by focusing on signals that admit a good polynomial approximation. We emphasize that these are only bounds, as 1) polynomials are only a special case of exponential polynomials and 2) the latter are only a special case of functions that yield a low-rank matrix $\mathbf{S}$. As such, assume that we approximate the underlying function $f(t)$ of $\mathbf{f}$ with a Taylor polynomial $p(t)$ of degree $R-1$ around $t = t_*$. Assuming $f(t)$ and its derivatives up to order R are continuous, which is satisfied for smooth signals, Taylor's theorem provides the following element-wise upper bound on the error in (4.9):

$$|f(t) - p(t)| \leq \frac{f_{\max}}{R!}|t - t_*|^R \tag{4.10}$$

with $f_{\max} = \max_u f^{(R)}(u)$, $u \in (t_*, t)$ and $f^{(R)}$ the Rth derivative of f. The corresponding matrix $\tilde{\mathbf{S}}$ of $p(t)$ has rank R, see Table 4.1; hence, (4.10) is a bound on the error of the rank-R approximation of $\mathbf{S}$. Signals with rapidly converging Taylor series admit an approximate low-rank model, hence, only a small R is needed for a good approximation. A general polynomial approximation $p(t)$ in K uniformly sampled points in the interval $[a, b]$ gives the following upper bound on (4.9):

$$\|\mathbf{f} - \text{vec}(\tilde{\mathbf{S}})\|_{\text{F}}^2 \leq \left(\frac{h^R}{4R}\bar{f}\right)^2$$

with $h = (b-a)/K$, $\bar{f} = \max_u f^{(R)}(u)$, $u \in [a, b]$, and $f^{(R)}$ the Rth derivative of f. Similar results can be derived for other types of approximations, e.g., a polynomial approximation in Chebyshev points. In Section 4.5, we illustrate our strategy for real-life signals as well, showing that our working hypothesis is valid for a variety of signals and applications.

In this chapter we also reshape signals into higher-order tensors, going further than the Hankel strategy from [101] and enabling an even more compact representation. In tensor-based scientific computing one often reshapes a

function up to a $(2 \times 2 \times \cdots \times 2)$ tensor of very high order to achieve maximal compression for a fixed rank R [171, 215]. Here, we allow much more freedom in the choice of the reshaping parameters, which enables a trade-off between the approximation error in (4.9) and the compression rate, see Subsection 4.5.5.

Let us now describe the strategy more formally. Suppose one reshapes the rth source $\mathbf{s}_r$ in (4.6) into a $(I \times J)$ matrix $\mathbf{S}_r$ such that $\text{vec}(\mathbf{S}_r) = \mathbf{s}_r$ with $K = IJ$. Note that this is the same as stacking different decimated versions of the signal in the rows of a matrix. If the rth reshaped (or matricized) source $\mathbf{S}_r$ admits a rank-1 representation, which is our working hypothesis, we have that $\mathbf{S}_r = \mathbf{a}_r \otimes \mathbf{b}_r$ with $\mathbf{a}_r \in \mathbb{K}^I$ and $\mathbf{b}_r \in \mathbb{K}^J$, as, e.g., in (4.7). In general, however, this model is too restrictive. The reshaped sources may admit, or better be approximated by a low-rank representation, as is, e.g., the case for a sine and the functions in Figure 4.2, respectively. Hence, we have that $\mathbf{S}_r = \sum_{l_r=1}^{L_r} \mathbf{a}_{l_r r} \otimes \mathbf{b}_{l_r r}$. Note that this means that we assume that the sources can be written as a sum of Kronecker products: $\mathbf{s}_r = \text{vec}\,(\mathbf{S}_r) = \sum_{l_r=1}^{L_r} \mathbf{b}_{l_r r} \otimes \mathbf{a}_{l_r r}$. This strategy enables a compact representation of the sources, see Table 4.2. Indeed, the number of parameters is one order of magnitude lower than the finite length K if $I \approx J$.

More generally, we can reshape the sources into a higher-order tensor, enabling a more compact representation. Suppose we reshape the rth source $\mathbf{s}_r$ into an Nth-order tensor $\mathcal{S}_r \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ such that $\text{vec}(\mathcal{S}_r) = \mathbf{s}_r$ with $K = \prod_{n=1}^{N} I_n$. If the rth reshaped (or tensorized) source $\mathcal{S}_r$ admits a (higher-order) low-rank representation, we have that:

$$\mathcal{S}_r = \sum_{l_r=1}^{L_r} \mathbf{u}_{l_r r}^{(1)} \otimes \mathbf{u}_{l_r r}^{(2)} \otimes \cdots \otimes \mathbf{u}_{l_r r}^{(N)}, \tag{4.11}$$

in which $\mathbf{u}_{l_r r}^{(n)} \in \mathbb{K}^{I_n}$ for $n = 1, \ldots, N$, where the number of rank-1 terms L_r can differ between sources. Note that this is a PD as in (4.1). This means that the sources can be modeled, or approximated, by sums of $(N-1)$ Kronecker products [171]:

$$\mathbf{s}_r = \text{vec}\,(\mathcal{S}_r) = \sum_{l_r=1}^{L_r} \mathbf{u}_{l_r r}^{(N)} \otimes \mathbf{u}_{l_r r}^{(N-1)} \otimes \cdots \otimes \mathbf{u}_{l_r r}^{(1)}, \tag{4.12}$$

In general, the number of parameter decreases logarithmically in the number of Kronecker products N (i.e., the order of the representation) and increases proportionally with the number of rank-1 terms L_r, see Table 4.2. For example, if $I_n = I$ for $n = 1, \ldots, 3$, then $K = I^3$ and only $O(3L_r I)$ parameters are needed. The possibly large compressions indicate the applicability of this strategy for large-scale BSS problems.

Table 4.2: Reshaping $\mathbf{s}_r$ in (4.6) into $\mathcal{S}_r \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ and then using a rank-L_r representation leads to a considerable compression. If $N = 2$, we use I and J. The number of parameters decreases logarithmically in N and increases proportionally with L_r.

	K	for general I_n	$I_n \approx I$, for all n
$N = 2$	IJ	$L_r(I + J - 1)$	$O(L_r I)$
$N > 2$	$\prod_{n=1}^{N} I_n$	$L_r(\sum_{n=1}^{N} I_n - N + 1)$	$O(L_r N I)$

4.2.3 Decomposition

We now demonstrate how the BSS problem in (4.6) can be reformulated as the computation of a tensor decomposition when the sources admit a low-rank representation. Let us start as follows: each row of $\mathbf{X}$ is reshaped into a $(I \times J)$ matrix as described earlier and then stacked into a third-order tensor $\mathcal{X} \in \mathbb{K}^{I \times J \times M}$ such that $\text{vec}\,(\mathbf{X}_m) = \mathbf{x}_m$. In other words, the mth matricized observed signal is equal to the mth frontal slice of $\mathcal{X}$. Since the tensorization is a linear operation, the M reshaped observed signals are linear combinations of the R reshaped sources $\mathbf{S}_r \in \mathbb{K}^{I \times J}$. As such, we have that:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{S}_r \otimes \mathbf{m}_r. \tag{4.13}$$

We denote this deterministic tensorization technique by *segmentation*; see Figure 4.1 for an illustration. Now assume that the rth reshaped source in (4.13) admits a rank-1 representation, i.e., $\mathbf{S}_r = \mathbf{a}_r \otimes \mathbf{b}_r$ for $r = 1, \ldots, R$, then we have that:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{m}_r. \tag{4.14}$$

Equation (4.14) is a CPD as defined in (4.1). Consequently, the BSS problem boils down to the computation of a CPD of a third-order tensor in R rank-1 terms. Analogously, if the reshaped sources admit a low-rank representation, the BSS problem boils down to a BTD in multilinear rank-$(L_r, L_r, 1)$ terms, as in (4.5) and illustrated in Figure 4.1. References to uniqueness results for both cases have been mentioned in Subsection 4.1.2. We insist that the compressibility of the sources has enabled their blind separation.

More generally, we can reshape each observed signal into a $(I_1 \times I_2 \times \cdots \times I_N)$ Nth-order tensor as described earlier and then stack it into a $(N + 1)$th-order tensor $\mathcal{X} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N \times M}$. As such, the mth tensorized observed signal is

equal to the mth Nth-order "frontal slice" of $\mathcal{X}$:

$$\mathcal{X} = \sum_{r=1}^{R} \mathcal{S}_r \otimes \mathbf{m}_r, \tag{4.15}$$

If the reshaped sources $\mathcal{S}_r \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ allow a low-rank representation as in (4.11), we have:

$$\mathcal{X} = \sum_{r=1}^{R} \left(\sum_{l_r=1}^{L_r} \mathbf{u}_{l_r r}^{(1)} \otimes \mathbf{u}_{l_r r}^{(2)} \otimes \cdots \otimes \mathbf{u}_{l_r r}^{(N)} \right) \otimes \mathbf{m}_r, \tag{4.16}$$

which is a decomposition in R (rank-L_r $\otimes$ vector) terms [332]. It is a more general decomposition because it boils down to a CPD of a higher-order tensor as in (4.1) if $L_r = 1$ for all r. Also, it boils down to a BTD in multilinear rank-$(L_r, L_r, 1)$ terms as in (4.5) if $N = 2$, i.e., if $\mathcal{X}$ is a third-order tensor. In that case, the factor matrices $\mathbf{U}_r^{(1)}$ and $\mathbf{U}_r^{(2)}$ of the rth term are not unique, but their products are (up to scaling and permutation). On the other hand, for $N > 2$, the factor matrices $\mathbf{U}_r^{(n)}$ are unique under mild conditions because they form a rank-L_r PD of an Nth-order tensor. We will exploit this in the DOA estimation application in Subsection 4.5.7.

The proposed method simultaneously determines both the mixing vectors *and* the sources by 1) simply reshaping the data (using segmentation) and 2) exploiting the fact that many real-life signals admit a (higher-order) low-rank representation. As such, the BSS problem boils down to a tensor decomposition and 3) we can benefit from mild uniqueness properties. Moreover, 4) it is applicable for large-scale BSS problems, i.e., large K, as is clear from the possibly huge compressions as indicated above. However, this is not necessarily a significant advantage compared to existing methods like ICA. The latter has only a linear dependence on K and even benefits from large K accuracy-wise because the K samples are used to estimate statistics. Finally, 5) the method is deterministic, meaning that it does not use (higher-order) statistics, hence, it also works well if the number of samples is small and/or if the sources are not statistically independent. This is a difference with statistical methods such as ICA.

4.3 Large-scale blind source separation via low-rank mixing vectors

In the previous section we exploited the fact that many real-life (source) signals admit a low-rank representation. This is also a natural assumption for the

mixing vectors if one considers, e.g., many sensors and/or high sensor density; we call them *low-rank mixing vectors* analogous to low-rank sources. Such problems arise in biomedical sciences, e.g., wireless body area networks (WBANs) using electroencephalography (EEG) [43] and electrocorticography (ECoG) [310] with high spatial resolution, or neural dust with thousands of miniature sensors (neural probes) dispersed throughout the brain [314]. Moreover, one often encounters mixing matrices with Vandermonde structure [337], i.e., each reshaped mixing vector has exactly rank one. An example are uniform linear (ULAs) and rectangular arrays (URAs) with far-field sources that emit narrowband signals [228, 268, 323]. Here, we also see a trend towards large-scale antennas, also known as massive MIMO [156, 236]. If the signals propagate through several distinct paths, e.g., due to reflections or scattering [202], each reshaped mixing vector has low rank. If the sources are located in the near-field, the Vandermonde structure is only approximate which can be accommodated by a low-rank approximation.

Exploitation of the underlying compactness of such low-rank mixing vectors amounts to a comparable method as in Section 4.2, which has been briefly addressed in [50]. Let us illustrate the analogy with the previous section more clearly: each *column* (cf. above) of $\mathbf{X}$ is reshaped into a $(I \times J)$ matrix with $M = IJ$ and then stacked into a third-order tensor $\mathcal{X} \in \mathbb{K}^{I \times J \times K}$. Next, assume the reshaped mixing vectors admit a rank-1 representation, which is our working hypothesis, i.e., $\mathbf{M}_r = \text{unvec}(\mathbf{m}_r) = \mathbf{a}_r \otimes \mathbf{b}_r$ for $r = 1, \ldots, R$. Hence,

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{s}_r. \tag{4.17}$$

Note that this boils down to applying the same strategy as before on the transposed observed data matrix. The generalization to higher-order low-rank representations is straightforward. The same analysis as in Subsection 4.2.3 applies, but now we segment the mixing vectors and exploit the fact that they possibly admit a (higher-order) low-rank representation. Moreover, the method has several advantages over ICA: ICA methods based on (full) HOS are infeasible when M is large as the number of entries in Qth-order statistics is $O(M^Q)$. Also, our method can handle Gaussian random sources in contrast to ICA (if the mixing vectors indeed exhibit some low-rank structure) [89]. Finally, the method imposes only mild conditions (via the uniqueness conditions) on the sources in contrast to existing methods, e.g., linear independence instead of statistical independence as in ICA.

4.4 Large-scale blind source separation using twofold segmentation

In the previous two sections we either reshaped the sources *or* the mixing vectors and then exploited the hypothesized low-rank structure. However, as we have illustrated before, both the mixing vectors *and* the sources may admit such a higher-order low-rank representation. Hence, a natural extension is to use both strategies simultaneously. For instance, one often has sinusoidal sources, which admit a rank-2 representation, in ULAs of which the Vandermonde mixing vectors admit a rank-1 representation. To the best of our knowledge, this is the first time that tensorization is used on both levels of the BSS problem and more generally in matrix factorization.

By exploiting the underlying compactness on both levels, we are again able to reformulate the BSS problem as the computation of a tensor decomposition. Let us start with reshaping each column of $\mathbf{X}$ into a $(I_1 \times I_2)$ matrix with $M = I_1 I_2$ and stacking them in an intermediate third-order tensor $\mathcal{Y} \in \mathbb{K}^{I_1 \times I_2 \times K}$. Note that the (i_1, i_2)th mode-3 vector of $\mathcal{Y}$ equals the $(i_1 + (i_2 - 1)I_1)$th row of $\mathbf{X}$. Each mode-3 vector of $\mathcal{Y}$ (i.e., row of $\mathbf{X}$) is subsequently reshaped into a $(J_1 \times J_2)$ matrix with $K = J_1 J_2$, which overall yields a fourth-order tensor $\mathcal{X} \in \mathbb{K}^{I_1 \times I_2 \times J_1 \times J_2}$. Hence, we have that:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{M}_r \otimes \mathbf{S}_r. \tag{4.18}$$

We denote this by *twofold segmentation* (cf. Sections 4.2 and 4.3). Let us now assume that both the reshaped mixing vectors and sources admit a rank-1 representation. In that case, it is easy to see that (4.18) is a CPD of a fourth-order tensor in R rank-1 terms. More generally, if the segmented mixing vectors and sources allow a low-rank representation, we have:

$$\mathcal{X} = \sum_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^{\mathrm{T}}) \otimes (\mathbf{C}_r \mathbf{D}_r^{\mathrm{T}}), \tag{4.19}$$

in which $\mathbf{A}_r \in \mathbb{K}^{I_1 \times L_r}$ and $\mathbf{B}_r \in \mathbb{K}^{I_2 \times L_r}$ have full column rank L_r and $\mathbf{C}_r \in \mathbb{K}^{J_1 \times P_r}$ and $\mathbf{D}_r \in \mathbb{K}^{J_2 \times P_r}$ have full column rank P_r. Note that the ranks L_r and P_r can be different for each r and do not necessarily have the same value inside the rth term. This is a new kind of decomposition: $\mathcal{X}$ is decomposed in a sum of R (rank-L_r $\otimes$ rank-P_r) terms.

More generally, we can reshape each row and column of $\mathbf{X}$ into $\mathcal{S}_r \in \mathbb{K}^{J_1 \times J_2 \times \cdots \times J_{N_s}}$ and $\mathcal{M}_r \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_{N_m}}$ such that $\mathrm{vec}(\mathcal{M}_r) = \mathbf{m}_r$ and

$\text{vec}(\mathcal{S}_r) = \mathbf{s}_r$, respectively, with $M = \prod_{n_m=1}^{N_m} I_{n_m}$ and $K = \prod_{n_s=1}^{N_s} J_{n_s}$, analogous to the single segmentation case in (4.15). As such, we have that:

$$\mathcal{X} = \sum_{r=1}^{R} \mathcal{M}_r \otimes \mathcal{S}_r.$$

Analogous to (4.16), the reshaped mixing vectors and sources can both admit a low-rank representation. Hence, we have that:

$$\mathcal{X} = \sum_{r=1}^{R} \left(\sum_{l_r=1}^{L_r} \otimes_{n_m=1}^{N_m} \mathbf{u}_{l_r r}^{(n_m)} \right) \otimes \left(\sum_{p_r=1}^{P_r} \otimes_{n_s=1}^{N_s} \mathbf{v}_{l_r r}^{(n_s)} \right),$$

in which $\mathbf{u}_{l_r r}^{(n_m)} \in \mathbb{K}^{I_{n_m}}$ and $\mathbf{v}_{p_r r}^{(n_s)} \in \mathbb{K}^{J_{n_s}}$. In comparison with (4.19), the block factors $\mathbf{U}_r^{(n)}$ and/or $\mathbf{V}_r^{(n)}$ are unique under mild conditions if $N_m > 2$ and/or $N_s > 2$. The reason is the same as for the single segmentation case, see Subsection 4.2.3.

The proposed method offers 1) a framework to exploit the low-rank structure of both the reshaped mixing vectors and sources; the same analysis as in the previous sections applies. Again, we reformulate the BSS problem as the computation of a tensor decomposition, hence, 2) we can benefit from the mild uniqueness properties. More specifically, it boils down to the computation of a new and more general decomposition. As such, 3) the method is applicable in a big data setting: it can handle both large sample sizes and large numbers of sensors efficiently, see Table 4.2. Furthermore, 4) the method is deterministic, hence, it is not needed *per se* to have a large number of samples. Finally, 5) only mild, and natural, assumptions are imposed on the mixing vectors and the sources. We simply exploit the low-rank structure which is often present in real-life signals as explained above.

4.5 Simulations and applications

In Subsection 4.5.1, we give an example of the separation of two low-rank sources and the separation of two low-rank sources that are mixed with low-rank mixing vectors. Subsection 4.5.2 demonstrates the separation of more sources than observed signals. We investigate the influence of the noise and the sample size in Subsection 4.5.3. Subsection 4.5.4 shows how well one can approximate the reshaped mixing vectors and/or sources for varying rank and SNR. In Subsection 4.5.5 we analyze the influence of the choice of reshaping dimensions. Finally, in the last two subsections, we illustrate the proposed methods with fetal electrocardiogram extraction and direction-of-arrival estimation in large-scale uniform linear arrays.

We use the `segmentize` command from Tensorlab to apply segmentation to the observed data matrices [392]. The CPD and BTD in multilinear rank-$(L_r, L_r, 1)$ terms can typically be computed algebraically by means of a generalized eigenvalue decomposition [104, 332, 333]. The algebraic solution is exact in the noiseless case and a good initialization for optimization-based methods in the noisy case. In this chapter, we use least squares optimization-based algorithms `cpd` and `lll` to fit the decomposition to the data until a sufficiently high accuracy is attained. During the computation, it is theoretically possible that degeneracy occurs [227, 327]. For example, the magnitude of some terms grows without bounds but with opposite sign, resulting in a poor solution but a good fit. Degeneracy can be avoided in several ways such as increasing the number of rank-1 terms or imposing orthogonality or nonnegative constraints on the factor matrices [84, 227, 248, 344]. The decompositions in (rank-$L_r \otimes$ vector) and (rank-$L_r \otimes$ rank-P_r) terms are computed with two adapted versions of `cpd_nls` called `lvec_nls` and `lp_nls`, respectively, and are available upon request. For very large tensors, one can resort to large-scale algorithms as described in [321, 391, 393].

The mixing vectors and sources can only be determined up to scaling and permutation, i.e., the standard indeterminacies in BSS. Hence, in order to compute the error they are first optimally scaled and permuted with respect to the true ones. The relative error is then defined as the relative difference in Frobenius norm, i.e., we have relative error $\epsilon_{\mathbf{A}} = ||\mathbf{A} - \hat{\mathbf{A}}||_{\mathrm{F}}/||\mathbf{A}||_{\mathrm{F}}$ with $\hat{\mathbf{A}}$ an optimally scaled and permuted estimate of $\mathbf{A}$. We use Gaussian additive noise unless indicated otherwise.

4.5.1 General experiments

First, we illustrate the method proposed in Section 4.2. Consider $R = 2$ low-rank sources: $s_1(t) = e^{-t}$ and $s_2(t) = \sin(4\pi t)$ with $K = 4096$ equidistant samples in $[0, 1]$. They are mixed into $M = 3$ observed signals using $\mathbf{M} = [0.5,\ 2;\ 2,\ -3;\ 1,\ 0.5]$. We use a second-order ($N = 2$) rank-1 ($L_1 = 1$) and rank-2 ($L_2 = 2$) approximation for the first and second source, respectively, with $I = J = 64$. Note that the approximation of the first and second source requires only 127 and 254 values, respectively, see Table 4.2. This is the maximal reduction for a second-order approximation. Namely, we have a compression of $1 - L_r \frac{I+J-N+1}{M}$, i.e., 96.90% and 93.80% for the first and second source, respectively. The perfectly recovered sources are shown in Figure 4.3.

Second, we illustrate the method proposed in Section 4.4. Consider $R = 2$ low-rank sources: $s_1(t) = e^{-t} + e^{t} - e^{0.5t}$ and $s_2(t) = 2e^{-t}$ with $K = 4096$ equidistant samples in $[0, 1]$. The sources are mixed with two low-rank mixing vectors:

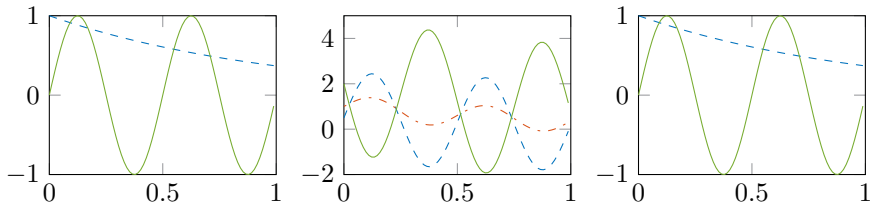


Figure 4.3: Results for the first experiment of Subsection 4.5.1. Left: the two original sources. Middle: the observed signals. Right: the recovered sources.

$m_1(\xi) = \sin(2\pi\xi)$ and $m_2(\xi) = e^{-2\xi}\sin(6\pi\xi)$ with $M = 4096$ equidistant samples in $[0, 1]$. We use a third-order ($N_s = 3$) rank-3 ($P_1 = 3$) and rank-1 ($P_2 = 1$) approximation for the first and second source, respectively, with $J_1 = J_2 = J_3 = 16$. Furthermore, we use a second-order ($N_m = 2$) rank-2 approximation for both mixing vectors ($L_1 = L_2 = 2$) with a non-optimal choice of the segmentation parameters: $I_1 = 128$ and $I_2 = 32$. Hence, we decompose the $(128 \times 32 \times 16 \times 16 \times 16)$ segmented version of $\mathbf{X}$ into a sum of a (rank-2 $\otimes$ rank-3) and a (rank-2 $\otimes$ rank-1) term. The approximation of the rth mixing vector requires only $L_r(I_1 + I_2 - N_m + 1)$ values, i.e., a compression of $1 - L_r\frac{I_1+I_2-N_m+1}{M} = 92.19\%$, although this is not the maximal compression. Higher compression can be attained by increasing the order. For instance, the approximation of the rth source consists of only $P_r(J_1 + J_2 + J_3 - N_s + 1)$ values, i.e., a compression of $1 - P_r\frac{J_1+J_2+J_3-N_s+1}{M}$. Specifically, we have a compression of 96.63% and 98.88% for the first and second source, respectively. We further investigate the choice of I_{n_m} and J_{n_s} in Subsection 4.5.5. The perfectly recovered factors are shown in Figure 4.4.

4.5.2 Underdetermined mixture

We illustrate the separation of more sources than observed signals. Consider $R = 3$ complex exponential source signals $s_r(t) = e^{2\pi irt}$ for $r = 1, \ldots, R$ which are mixed into $M = 2$ observed signals using $\mathbf{M} = [-1,\ 0.5,\ 2;\ 0.5,\ 1,\ 0.5]$. We take $K = 4096$ uniformly discretized samples in $[0, 1]$. We use a second-order ($N = 2$) rank-1 approximation for both sources with $I_1 = I_2 = 64$. The real part of the recovered sources is shown in Figure 4.5: perfect reconstruction is obtained.

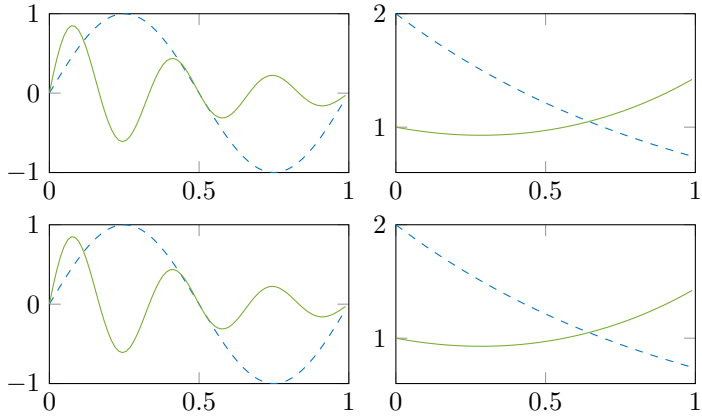


Figure 4.4: Results for the second experiment of Subsection 4.5.1. Top: original mixing vectors (left) and sources (right). Bottom: perfectly recovered mixing vectors (left) and sources (right).

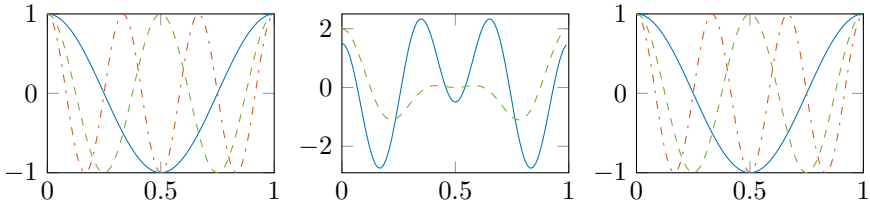


Figure 4.5: Results for the underdetermined mixture. The real part of the three original sources (left), the two observed signals (middle), and the recovered sources (right).

4.5.3 Noise and sample length

First, we investigate the influence of the noise and the sample size K for the method of Section 4.3. Consider a setup in which we have $M = 4096$ sensors and $R = 2$ i.i.d. zero-mean unit-variance Gaussian random sources of length $K = \{10^1, 10^2, 10^3\}$. We construct the low-rank mixing vectors as the vectorization of a second-order ($N = 2$) rank-2 ($L_1 = 2$) and rank-3 ($L_2 = 3$) tensor using (4.12) with zero-mean unit-variance Gaussian random factor vectors and $I = J = 64$. Hence, we use a second-order rank-2 and rank-3 approximation with $I = J = 64$, respectively. In Figure 4.6, we report the relative error on the mixing vectors $\epsilon_{\mathbf{M}}$ and the sources $\epsilon_{\mathbf{S}}$; note that the results are very accurate in comparison with the SNR. Although the method is deterministic, it

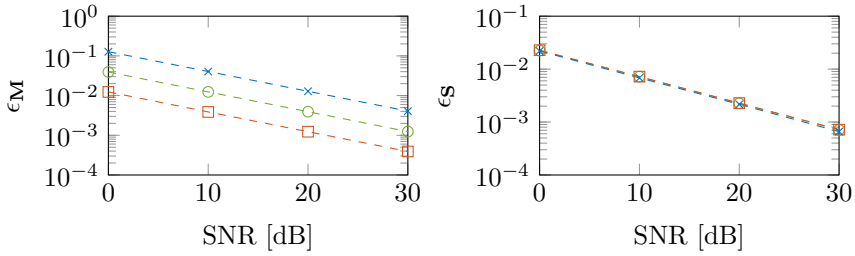


Figure 4.6: Median across 100 experiments of the relative error on the mixing vectors (left) and the sources (right) as a function of SNR for $K = 10^1$ (- ⋆ -), 10^2 (- ⊖ -), and 10^3 (- ⊡ -). The mixing vectors are well conditioned.

is beneficial to increase K under noisy conditions. However, K can be (very) low in comparison to typical values in ICA. (Note that in this particular example, ICA cannot be used since the sources are Gaussian.) $\epsilon_{\mathbf{S}}$ does not improve for increasing K because one also has to estimate longer source signals. Similar results can be obtained for the method of Section 4.2 when increasing the number of sensors M under noisy conditions.

Next, consider a similar setup as in the previous experiment but now with the following rank-1 mixing vectors: $m_1(\xi) = e^{0.5\xi}$ and $m_2(\xi) = e^{-2\xi}$ with $\xi \in [0, 1]$. We use a second-order ($N = 2$) rank-1 approximation for both mixing vectors ($L_1 = L_2 = 1$) with $I = J = 64$. The results are shown in Figure 4.7: in comparison with Figure 4.6, there is some loss of accuracy on the mixing vectors and much clearer on the sources. This is due to the condition of the problem: in the previous experiment, the mixing vectors are approximately orthogonal and have about the same size ($\|\mathbf{m}_1\|/\|\mathbf{m}_2\| \approx 0.8$), while now the angle is 37.11° and $\|\mathbf{m}_1\|/\|\mathbf{m}_2\| = 2.65$. Hence, the computation of the decomposition is more difficult and the estimates less accurate.

4.5.4 Low-rank approximation

We investigate the influence of deviations from a second-order rank-1 structure on the relative error as follows. Define each mixing vector as the vectorization of a random matrix with exponentially decaying singular values, i.e., $\mathbf{m_r} = \text{vec}\left(\mathbf{U_r}\text{diag}\left(\boldsymbol{\sigma}\right)\mathbf{V_r}\right)$ with $\boldsymbol{\sigma} = e^{-\alpha\boldsymbol{\xi}}$ and $\boldsymbol{\xi}$ a vector containing $\min\left(I, J\right)$ equidistant samples in $[0, 1]$. $\mathbf{U}_r$ and $\mathbf{V}_r$ are random orthogonal matrices of compatible dimensions. The exponential decay of the singular values is controlled with α which is a measure for the rank-1-ness of the mixing vectors: increasing α leads to more rank-1-like mixing vectors and vice-versa. We take

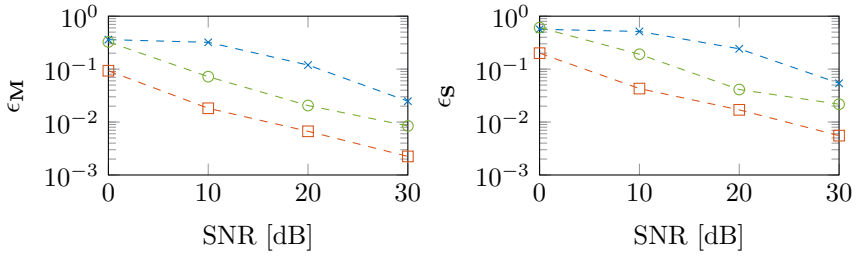


Figure 4.7: Median across 100 experiments of the relative error on the mixing vectors (left) and the sources (right) as a function of SNR for $K = 10^1$ (- ⋆ -), 10^2 (- ⊖ -), and 10^3 (- ⊡ -). The mixing vectors are ill-conditioned.

$R = 2$ i.i.d. zero-mean unit-variance Gaussian random sources of length $K = 10$ and use a second-order ($N = 2$) rank-L_r approximation with $I = J = 64$.

Figure 4.8 shows the relative errors $\epsilon_{\mathbf{M}}$ and $\epsilon_{\mathbf{S}}$ as a function of α for an SNR of 15 dB and 25 dB using $L_1 = L_2 = 1$. Note that an estimate of the mixing matrix $\hat{\mathbf{M}}$ can be obtained from the decomposition, i.e., from (4.17) for this particular case, in the way explained above. However, one can also estimate it via the noisy observed data matrix and the pseudo-inverse of the estimated source matrix: $\hat{\mathbf{M}} = \mathbf{X}\hat{\mathbf{S}}^{\dagger}$. The figure illustrates that $\epsilon_{\mathbf{M}}$ decreases for increasing α until it stagnates due to noise. One can also see that, for large α, $\hat{\mathbf{M}}$ computed via the pseudo-inverse is less accurate than directly extracting $\hat{\mathbf{M}}$ from (4.17) and imposing rank-1 structure. However, for small α, the opposite is true. Indeed, for decreasing α, the mixing vectors become less rank-1 like and our rank-1 model cannot attain a better estimate than the one given by Eckart–Young's theorem [142]. Also, note that the sources are estimated more accurately than the mixing vectors: the noise on the sources is more averaged out because this factor is much shorter in the decomposition ($K \ll I, J$) [115].

Figure 4.9 shows the relative errors for several choices of L_r. One can observe that for increasing L_r, the relative error decreases in the case of small α, i.e., in the case of little rank-1-like mixing vectors. On the other hand, little is lost through overmodeling (i.e., choosing L_r too large) for large α. In fact, we overmodel less than conventional methods as we exploit the low-rank structure. Hence, the choice of L_r is not so critical, see [101, 124]. In this case one also knows that the multilinear rank of $\mathcal{X}$ is bounded by $(\sum_{r=1}^{R} L_r, \sum_{r=1}^{R} L_r, R)$.

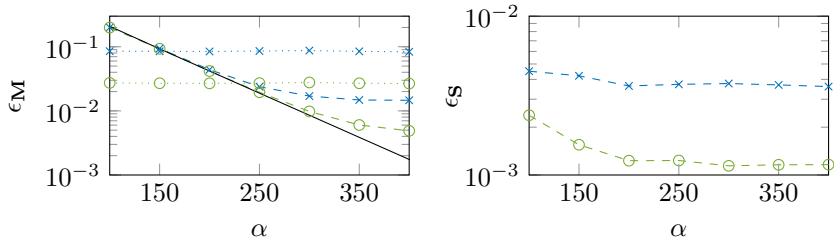


Figure 4.8: Median across 100 experiments of the relative error on the mixing vectors (left), extracted from (4.17) (dashed) and computed via the inverse of $\hat{\mathbf{S}}$ (dotted), and the sources (right) for varying rank-1-ness α and an SNR of 15 dB (cross) and 25 dB (circle). The error bound given by the Eckart-Young theorem is shown in solid.

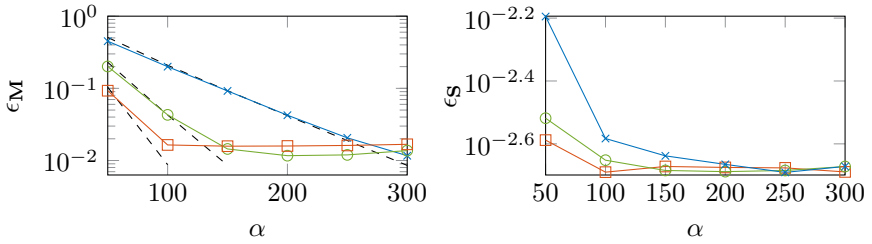


Figure 4.9: Median across 100 experiments of the relative error on the mixing vectors (left) and the sources (right) for varying rank-1-ness α of the mixing vectors and 20 dB SNR with $L_1 = L_2 = 1$ (—×—), $L_1 = L_2 = 2$ (—○—), and $L_1 = L_2 = 3$ (—□—). The error estimate given by the Eckart-Young theorem is shown with dashed lines. Note the small scale of $\epsilon_{\mathbf{S}}$ (right).

4.5.5 Compression versus accuracy

We investigate the trade-off between compression and accuracy which will lead to a better understanding on how to choose the segmentation parameters I_{n_m} and/or I_{n_s}. We do this by examining the accuracy of a low-rank approximation of various segmentations of a real-life EEG signal with a sample rate of 500 Hz. More precisely, we reshape the EEG signal of length $K = 2^{14}$ into a $(I \times J)$ matrix with $I = 2^q$ and vary $q = 2, \ldots, 12$, then $J = 2^{14-q}$ such that $K = IJ$. Subsequently, we approximate the reshaped signal with a rank-L model with $L = \{1, 2, 3\}$.

In Figure 4.10, we plot the normalized number of parameters $\hat{K} = L(I + J)/K$ versus the relative error ϵ of the rank-L approximation. We see a clear trade-off

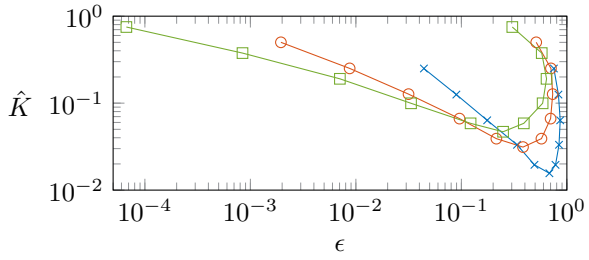


Figure 4.10: Normalized number of parameters $\hat{K}$ as a function of the relative error of a rank-1 (- ⋆ -), rank-2 (- ⊙ -), and rank-3 (- ⊡ -) approximation of a segmented real-life EEG signal of length $K = 2^{14}$. The signal is reshaped into a $(I \times J)$ matrix with $I = 2^q$ and $J = 2^{14-q}$ such that $K = IJ$ with $q = 2, \ldots, 12$ and q increasing from left to right on the curve.

between compression and accuracy, hence, what is considered a "good" choice of parameters will depend on the needs in a particular application. First of all, the curves are not symmetric since segmentation is not symmetric in the modes that it creates. Note that one can easily improve the accuracy without affecting the compression rate by switching the values of I and J such that $I < J$ rather than $I > J$ for the same rank. For fixed I and J, increasing the rank can greatly improve the accuracy, e.g., when $I \ll J$ (left part of Figure 4.10). The original signal and two particular approximations are shown in Figure 4.11. Note the relative error decreased from 0.68 to 0.096 by taking $I < J$ and increasing L for the second approximation. On the other hand, the compression reduced from 96.88% to 86.72%.

In general, a good choice of the parameters will depend on the application. If compression is the objective, one should choose $I \approx J$ and L not too large. If, on the other hand, accuracy is the objective, one can try other choices of I and J and maybe a higher rank L. In practice, one can try a particular choice of parameters, perform a similar analysis as here on the estimated sources, and further refine the choice from there.

4.5.6 Fetal electrocardiogram extraction

We use the method of Section 4.2 for the extraction of the antepartum fetal electrocardiogram (FECG) from multilead cutaneous (i.e., recorded on the mother's skin) potential recordings. The FECG is important for analyzing the health and condition of the fetus. The elimination of the mother's dominant heartbeat in the ECG can be seen as a BSS problem and one can use methods

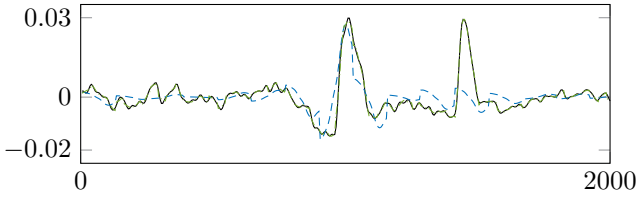


Figure 4.11: Visualization of the original EEG signal (——) and two approximations. The latter are obtained by first reshaping the original signal into a $(2^7 \times 2^7)$ and $(2^5 \times 2^9)$ matrix, respectively, and then approximating them by a rank-1 (- - - -) and rank-2 (-·-·-) matrix by truncating the singular value decomposition. The reconstructed signals are obtained by vectorizing these low-rank matrices. Only the first 2000 samples are shown. The rank-2 approximation is much better than the rank-1 as is also clear from Figure 4.10.

such as ICA [111]. ICA, however, falls short when only a few samples or heartbeats are available. FECG extraction is not a large-scale problem, but it is useful to illustrate a few features of our approach. Our method is applicable here because the typical QRS complexes in the ECG admit a low-rank approximation. In other words, we show that representability by a small number of parameters can be used as a ground for blind ECG signal separation. We illustrate our method for a real-life dataset.

The dataset contains eight observed signals, of which five abdominal and three thoracic; the dataset is available from DaISY[1]. Data acquisition and preprocessing is described in [65]. The sampling rate is 250 Hz. We only use the first 500 samples and scale each signal to unit norm. Each observed signal is segmented into a (25×20) matrix and the overall data set is stacked into a $(25 \times 20 \times 8)$ tensor. We use a rank-5 approximation for each source $(L_1 = L_2 = L_3 = L = 5)$. At least three sources are needed to extract the FECG; this is also the case for ICA [111]. We use this particular segmentation as to maximize the compression which is only an arbitrary choice. We determined L by a trial-and-error approach starting from a rank-10 approximation and then decreasing L. Little is lost by choosing a larger L anyway, see Subsection 4.5.4. Figure 4.12 shows two recovered sources. One can verify that the heartbeats of the fetus are no longer visible in the ECG of the mother and vice versa, i.e, we have a clear separation. The frequency of the FECG is typically twice as high as the frequency of the MECG, which can be observed as well.

[1] Available from `http://homes.esat.kuleuven.be/~smc/daisy/daisydata.html`

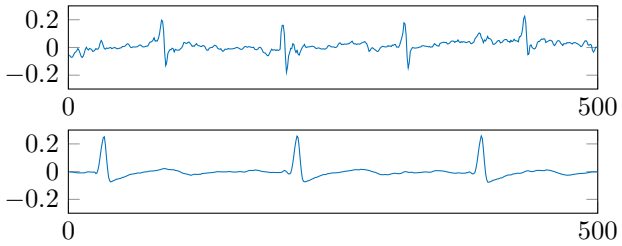


Figure 4.12: Visualization of the two recovered sources in the FECG experiment. Notice a clear separation of the fetal (above) and maternal (below) ECG.

4.5.7 Direction of arrival estimation

We use the method of Section 4.4 for direction-of-arrival (DOA) estimation of signals impinging on a ULA. Applications include radar, sonar, wireless communications, and seismic exploration. Recently, there has been a trend towards large-scale array processing [236]. Our method is able to cope with a large number of sensors, where other methods fall short. We compare our results with two well-known DOA estimation methods, MUSIC and ESPRIT, in several scenarios [228].

Consider a ULA that consists of M uniformly spaced and omnidirectional antennas receiving signals from R narrow-band sources located in the far field. In that case, the problem can be described by (4.6) with the mixing vectors defined element-wise as $m_{mr} = \theta_r^{m-1}$ with $\theta_r = e^{-2\pi i \Delta \sin(\alpha_r)\lambda^{-1}}$. Δ is the inter-element spacing, the angle α_r to the normal is the rth DOA (i.e., $-90° \leq \alpha_r \leq 90°$), and λ denotes the wavelength. Note that the mixing vectors are Vandermonde vectors: $\mathbf{m}_r = \begin{bmatrix} 1 & \theta_r & \theta_r^2 & \cdots & \theta_r^{M-1} \end{bmatrix}^{\mathrm{T}}$, hence, they admit a rank-1 representation [268, 322]. In a multipath setting, the mixing vectors are defined element-wise as $m_{mr} = \sum_{l=1}^{L_r} \theta_{lr}^{m-1}$ (ignoring path losses for simplicity), with L_r the number of paths for the rth source, admitting a low-rank representation. If the sources are located in the near field, the mixing vectors no longer admit a rank-1 representation but can still be well approximated by a low-rank model. If one also uses low-rank source models, we can use the method of Section 4.4.

First, consider a ULA with $M = 64$ sensors and Δ equal to halve the wavelength. Although our method is applicable for a large number of sensors, we choose M rather small so we can compare with MUSIC and ESPRIT. The latter two methods have to compute a $M \times M$ covariance matrix and then apply an eigenvalue decomposition (EVD). These steps can be computationally expensive because they have a complexity of $\mathcal{O}(M^2K)$ and $\mathcal{O}(M^3)$, respectively, rendering

such methods infeasible for large M and K. Moreover, MUSIC has to evaluate the MUSIC spectrum for many angles in order to estimate the DOAs accurately. Here, we evaluated the MUSIC spectrum in 10^4 equidistant angles in $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Note that the number of evaluation points bounds the attainable accuracy. Consider $R = 2$ low-rank sources: $s_r(t) = \sin(10\pi rt)$ with $K = 1024$ equidistant samples in $[0, 1]$. The sources are in line-of-sight and impinge on the ULA with $\alpha_{11} = 32°$ and $\alpha_{12} = 34°$. We use a second-order ($N_s = 2$) rank-2 ($P_1 = P_2 = 2$) approximation for both sources with $J_1 = J_2 = 32$ and a second-order ($N_m = 2$) rank-1 ($L_1 = L_2 = 1$) approximation for both mixing vectors with $I_1 = I_2 = 8$. Note that the model of the sources and mixing vectors requires only 126 and 15 values instead of 1024 and 64, respectively, see Table 4.2. This results in a compression of $1 - P_r \frac{J_1+J_2-N_s+1}{K} = 87.70\%$ and $1 - L_r \frac{I_1+I_2-N_m+1}{M} = 76.56\%$, respectively. In Figure 4.13 (left), we report the median of the relative errors on the DOAs ϵ_α. It is clear that the dedicated methods estimate the DOAs more accurately than our method. On the other hand, by exploiting the low-rank structure, we show that it is still possible to get fairly accurate estimates in comparison with well-known dedicated methods. Moreover, our method is applicable for large M.

In a second experiment, we add a third source ($R = 3$) that impinges on the ULA from two different paths ($L_3 = 2$): $\alpha_{13} = -15°$ and $\alpha_{23} = 67°$. We use a third-order ($N_m = 3$) rank-1 and rank-2 approximation for the first two and last mixing vector, respectively, with $I_1 = I_2 = I_3 = 4$. We choose $N_m > 2$ such that the different DOAs of the third source can be found directly from the estimated vectors $\mathbf{u}_{13}^{(1)}$ and $\mathbf{u}_{23}^{(1)}$ (instead of the column space of $\mathbf{S}_3$), see the discussion of uniqueness in Subsection 4.2.3. Note that one simply has to increase the rank L_r in order to cope with a multipath source, while MUSIC and ESPRIT need additional spatial smoothing [318]. The results are shown in Figure 4.13 (right).

In a third experiment, we use the same setup as in the first experiment but with two near-field sources defined by a DOA and range relative to the first antenna: $\alpha_1 = -17°$, $w_1 = 2(M-1)\Delta$, $\alpha_2 = 41°$, and $w_2 = 3(M-1)\Delta$. We compare our results with a two-dimensional version of MUSIC [197]. Figure 4.14 shows the median of the relative errors on the DOAs ϵ_α and the ranges ϵ_w. MUSIC estimates both the DOA and range more accurately but is even more computationally expensive because now one has to evaluate a two-dimensional spectrum for many angles and ranges. Here, we used 10^2 equidistant angles and ranges in $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and $[5, 12]$, respectively. In order to cope with near-field sources in our approach, one simply has to increase the rank L_r.

The final experiment uses the same setup as the first experiment but now with $M = 9$ and $K = 100$ with $J_1 = J_2 = 10$ and $I_1 = I_2 = 3$. As can be seen from

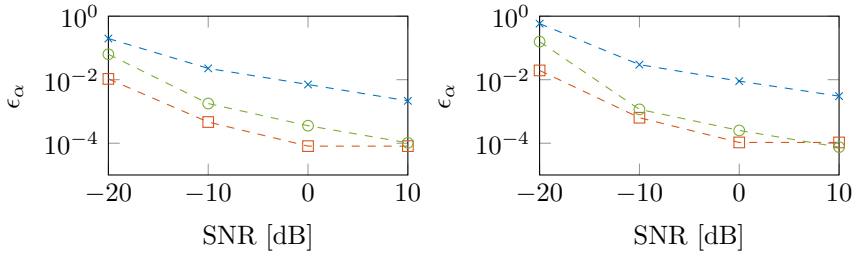


Figure 4.13: Median across 100 experiments of the relative error on the DOAs as a function of SNR for the line-of-sight (left) and multipath (right) experiment using segmentation (- ×-), ESPRIT (- ⊖-), and MUSIC (- ⊟-).

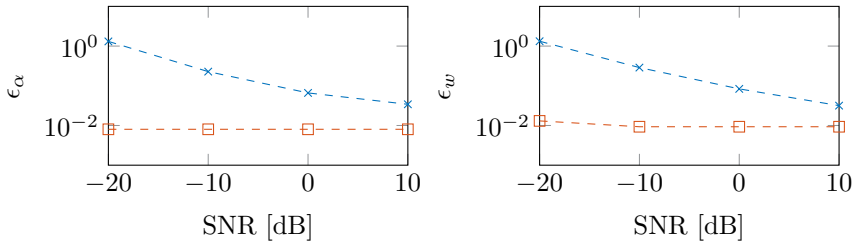


Figure 4.14: Median across 100 experiments of the relative error on the DOAs (left) and ranges (right) as a function of SNR for the near field experiment using segmentation (- ×-) and MUSIC (- ⊟-).

Figure 4.15, MUSIC fails to distinguish close DOAs when only a few samples are available and the SNR is low [228]. A small number of sensors M flattens the peaks in the MUSIC spectrum, making the problem more difficult. Our method can still estimate the DOAs accurately in such a setup because it is deterministic, performing even better than ESPRIT.

4.6 Conclusion

In this chapter, we have introduced a new method for BSS that exploits the fact that many real-life signals are compressible. We expressed this by assuming that the tensorized sources can be well approximated by a low-rank model. In other words, we assume that the sources can be well approximated by sums of Kronecker products of smaller vectors. As such, we have demonstrated that, if the sources indeed admit such a low-rank representation/approximation,

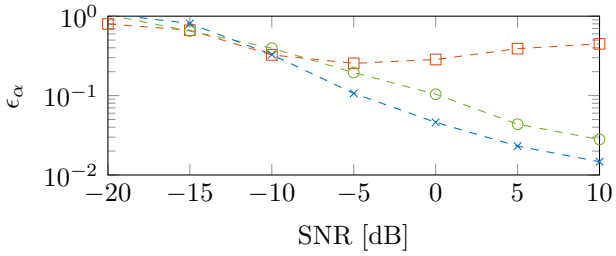


Figure 4.15: Median across 100 experiments of the relative error on the DOAs as a function of SNR for the small-scale line-of-sight experiment using segmentation (- ⋆ -), ESPRIT (- ⊖ -), and MUSIC (- ⊡ -).

the BSS problem boils down to the computation of a decomposition of the resulting tensorized observed data matrix. It is precisely the compressibility, which is essential in large-scale problems, that makes it very likely that the tensor decomposition is unique. Hence, our method provides a unique solution to the BSS problem and a way to cope with large-scale problems. Furthermore, we applied the same strategy to the mixing level motivated by an increasing number of sensors and sensor density in fields such as biomedical sciences and array processing. Moreover, combining both strategies simultaneously allowed the exploitation of low-rank structure on both levels of the BSS problem. We have illustrated our methods with two applications: FECG and DOA estimation for large-scale ULAs. We note that it is possible to impose constraints on the sources and/or mixture when applicable, e.g., statistical independence of the sources as in ICA. Such variants are out of the scope of this chapter. Although we focused on the CPD for modeling the tensorized sources and/or mixture, it is possible to consider other tensor models such as tensor trains (TTs) and hierarchical Tucker [171]. The latter are often used in tensor-based scientific computing because they combine large compression rates with good numerical properties. For the CPD of very large tensors, algorithms such as the ones in [321, 391, 393] can be used.

Chapter 5

Analytical multi-modulus algorithms based on coupled canonical polyadic decompositions

Abstract We present new techniques for multiple-input-multiple-output blind signal separation and blind system identification of multi-modulus source signals. Multi-modulus signals, such as 16-QAM, are very common in telecommunications. We algebraically transform the problem into a set of coupled tensor decompositions for which uniqueness results and algebraic solutions exist. An exact solution is guaranteed to be obtained by a matrix eigenvalue decomposition in the noiseless case. The proposed technique is deterministic and does not rely on statistics. Furthermore, we explain that certain source signals can still be recovered in the case of a rank-deficient mixing matrix. As a side result, we generalize a rank-1 detection procedure from a previously proposed tensor decomposition method. In the multi-modulus context, the generalization allows a reduction of the required number of samples for separation.

Reference This chapter is a slightly adapted version of the article [122]. Changes are limited to layout and representation aspects. The candidate performed the research and wrote the article under the guidance of the coauthors.

5.1 Introduction

Given a set of signals $\mathbf{X}$, the multiple-input-multiple-output (MIMO) blind signal separation (BSS) problem consists of the identification of the mixing matrix $\mathbf{M}$ and/or the original sources $\mathbf{S}$. In the instantaneous linear MIMO BSS model, one considers the following model with $\mathbf{N}$ the additive noise matrix:

$$\mathbf{X} = \mathbf{MS} + \mathbf{N}. \tag{5.1}$$

In blind system identification (BSI), an unknown system transfer function is considered instead of an instantaneous mixing matrix. In case of a linear time-invariant system, the observed signals are obtained through a convolution. Hence, the problem is also known as blind deconvolution.

For both BSS and BSI, solutions can be obtained under different working hypotheses. One hypothesis is mutual statistical independence of the source signals, leading to independent component analysis (ICA) [2, 89]. A deterministic hypothesis for multiple-invariance sensor array processing can be found in [323]; other deterministic assumptions are discussed in [119]. In telecommunications, blind separation and identification avoid the use of training sequences. A lot of work has been done in exploiting prior knowledge on the moduli of the samples of telecommunication signals. An example is the constant modulus algorithm (CMA), discussed in [164] and [363]. It was originally presented in a context of convolutive single-input-single-output (SISO). In CMA, all source samples are assumed to have the same constant modulus (CM), as is the case for real-valued binary phase shift keying (BPSK) signals, or complex-valued continuous phase/frequency shift keying (CPSK/CFSK) and complex 4-QAM (quadrature amplitude modulation) signals. Several important improvements followed in [166] and [37]. In [282], an algorithm for CMA was developed which was later coined as the multi-modulus algorithm [401, 402]. The algorithm applies a cost function with multiple linear moduli to fix the rotational ambiguity for 4-QAM signals. Despite the terminology, we will show that the method is not suitable for the problem discussed in this chapter. Various other cost functions have been analyzed for CMA, grouped in different families [3, 169, 245].

Instead of optimizing a certain CM cost function, the analytical constant modulus algorithm (ACMA) transforms the MIMO BSS problem analytically into a simultaneous matrix diagonalization [384]. This was one of the first appearances of the constant-modulus problem in a MIMO context. ACMA was later extended to the convolutive case [100, 387]. In [100], a connection was made to tensor algebra, as it was shown that ACMA boils down to the computation of a tensor decomposition of a partially Hermitian third-order tensor.

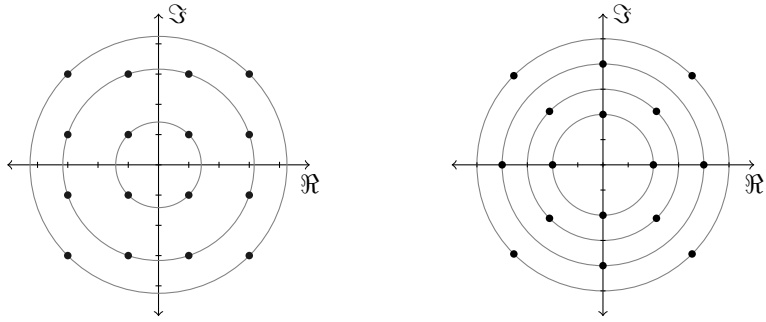


Figure 5.1: Constellation diagram for rectangular (left) and circular (right) 16-QAM, in which three and four squared moduli are present, respectively. CMA is only applicable when samples lie on a single circle. For rectangular 16-QAM, the squared moduli are 2, 10 and 18. For circular 16-QAM, the squared moduli are 4, 9, 16 and 25.

Many signals, such as higher-order QAM, do not satisfy the constant modulus property. One has to resort to more general techniques allowing signal samples with a broader range of possible moduli. Signals of which the sample moduli are not restricted to a single constant modulus but rather to a finite set of moduli are called multi-modulus signals. The samples then lie on a number of concentric circles. An example is a rectangular 16-QAM signal with samples drawn from $\pm\{1,3\} \pm \{1,3\}j$, having squared moduli of 2, 10 or 18. Circular 16-QAM involves four moduli. Both constellations are illustrated in Fig. 5.1.

It was shown in [315] that the behavior of CMA is poor for the multi-modulus MIMO BSS and BSI problem, especially when the source signals are correlated and nonuniform. Because the MMA algorithm from Oh [282] uses the dispersion of real and imaginary parts separately, it was presented as more suitable for higher-order QAM constellations [98, 187, 198]. Note that the definitions of the dispersion moduli used in the latter methods differ intrinsically from the definition of the moduli of multi-modulus signals. The appendix of this chapter contains a further clarification. Others applied Givens rotations on the MMA cost function [316] or introduced customizations to the default MMA algorithm [4].

This chapter proposes a new method for MIMO BSS and BSI of multi-modulus signals as a sound generalization of the ACMA algorithm. Unique to our MIMO approach is, first, that a more suitable cost function is used than the one in MMA. It is related to finite alphabet methods [244, 308, 414] and has been used in a convolutive SISO context in [275, 315].

Second, the problem is algebraically transformed into a set of coupled tensor decompositions. Coupled tensor decompositions, and more specifically coupled canonical polyadic decompositions (cCPDs), form an important and emerging concept in signal processing and data analytics. They enable the integration of data from multiple datasets, often called data fusion [6, 232, 333]. Using cCPDs, one-dimensional harmonic retrieval and shift-invariance techniques can be generalized to multi-dimensional harmonic retrieval and multiple-invariance techniques, respectively. Furthermore, one can deal with non-uniform linear arrays, incomplete arrays and convolutive mixtures instead of uniform linear arrays, dense arrays and instantaneous mixtures, respectively [342, 343].

Because of the reformulation into the tensor framework, algorithms and uniqueness conditions can be readily obtained [345, 346]. Algebraic methods using matrix eigenvalue decompositions are available such that an exact solution can be obtained in the noiseless case. In the presence of noise, refinement optimization techniques can be used. Furthermore, we present explicit lower bounds for the number of samples. Note that the method does not make use of statistics, and is therefore not dependent on the corresponding estimation error.

As a side result of this chapter, we generalize a rank-1 detection procedure from a previously proposed algebraic CPD algorithm [99]. While the original method finds Kronecker-structured vectors in a space spanned by a number of Kronecker-structured vectors, the generalization allows a search space spanned by both Kronecker-structured and arbitrary vectors. The generalized method is applied in the context of the multi-modulus technique to obtain a reduction of the number of samples required for separation.

The chapter is organized as follows. Multilinear algebra is introduced in Section 5.2. The BSS method is proposed in Section 5.3. An advanced version is discussed in Section 5.4 making use of a generalized rank-1 detection procedure. The methods are extended to BSI in Section 5.5. Simulations are presented in Section 5.6 and a discussion and conclusion follows in Section 5.7.

5.2 Multilinear algebra and notation

Tensors, denoted by calligraphic letters, e.g., $\mathcal{A}$, are higher-order generalizations of vectors (denoted by boldface lowercase letters, e.g., $\mathbf{a}$) and matrices (denoted by boldface uppercase letters, e.g., $\mathbf{A}$). Scalars are written as italic lowercase letters, e.g., a. The entry with row index i and column index j of a matrix $\mathbf{A} \in \mathbb{C}^{I \times J}$ is denoted by a_{ij}. Likewise, the $(i_1, i_2, \ldots, i_N)$th entry of an Nth-order tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ is denoted by $a_{i_1 i_2 \ldots i_N}$. The jth column of

a matrix $\mathbf{A} \in \mathbb{C}^{I \times J}$ is denoted by $\mathbf{a}_j$. The superscripts $\cdot^{\mathrm{T}}$, $\cdot^{\star}$, $\cdot^{\mathrm{H}}$, $\cdot^{-1}$ and $\cdot^{\dagger}$ represent the transpose, complex conjugate, complex conjugated transpose, inverse and Moore–Penrose pseudoinverse, respectively.

The Kronecker and column-wise Khatri–Rao products are denoted by $\otimes$ and $\odot$, respectively. The operation $\otimes$ stands for the outer product. The Kronecker and outer product are related, as for vectors $\mathbf{a} \in \mathbb{C}^I$ and $\mathbf{b} \in \mathbb{C}^J$ it holds that $\mathbf{a} \otimes \mathbf{b} = \mathrm{vec}(\mathbf{b} \otimes \mathbf{a})$, with vec the column-wise vectorization of a matrix or tensor. Concatenations along the first and second mode are denoted as $\left[\cdot\,;\,\cdot\right]$ and $\left[\cdot \quad \cdot\right]$, respectively.

A Kronecker-structured vector $\mathbf{v} \in \mathbb{C}^{IJ}$ is defined as a vector which can be written as a Kronecker product of two non-zero vectors $\mathbf{a} \in \mathbb{C}^I$ and $\mathbf{b} \in \mathbb{C}^J$ such that $\mathbf{v} = \mathbf{a} \otimes \mathbf{b}$. It can be seen that such a vector is a reshaped rank-1 matrix.

If an Nth-order tensor $\mathcal{A}$ can be written as an outer product of non-zero vectors, i.e., $\mathcal{A} = \mathbf{a}^{(1)} \otimes \mathbf{a}^{(2)} \otimes \cdots \otimes \mathbf{a}^{(N)}$, it has rank 1. If a tensor can be written as a sum of R rank-1 terms, the decomposition is called a polyadic decomposition (PD):

$$\mathcal{A} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \ldots \otimes \mathbf{a}_r^{(N)} \triangleq \left[\!\!\left[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)} \right]\!\!\right].$$

The factor vectors $\mathbf{a}_r^{(n)}$ are the columns of the factor matrices $\mathbf{A}^{(n)}$. If R is minimal, the decomposition is a canonical polyadic decomposition (CPD) and R is defined as the rank of the tensor. One advantage of the CPD is that it is unique under very mild conditions [135, 230]. There exists ample literature on tensors and tensor decompositions; we refer the interested reader to [84, 223, 324].

5.3 Tensor-based multi-modulus blind signal separation

We recall the blind signal separation problem:

$$\mathbf{X} = \mathbf{M}\mathbf{S} + \mathbf{N}.$$

Let us assume R sources, K observed signals and N samples for each signal; hence, $\mathbf{X}, \mathbf{N} \in \mathbb{C}^{K \times N}$, $\mathbf{M} \in \mathbb{C}^{K \times R}$ and $\mathbf{S} \in \mathbb{C}^{R \times N}$. We assume that $K = R$, as a preprocessing step based on principal component analysis (PCA) can be carried out when $K > R$. The underdetermined case is not discussed. Note that the distinction between K and R is still made to improve readability.

By defining the separation matrix $\mathbf{W} \in \mathbb{C}^{K \times R}$ as $\mathbf{W}^{\mathrm{T}} = \mathbf{M}^{\dagger}$ and by omitting $\mathbf{N}$ for convenience, one can write $\mathbf{S} = \mathbf{W}^{\mathrm{T}}\mathbf{X}$. The transpose is meant to simplify notation further on. Considering a source sample $s_{rn} = s_r(n)$, one can write

$$\forall r, n: \quad s_{rn} = \mathbf{w}_r^{\mathrm{T}} \mathbf{x}_n, \tag{5.2}$$

with $\mathbf{w}_r$ the rth column of $\mathbf{W}$ and $\mathbf{x}_n$ the nth column of $\mathbf{X}$.

Constant modulus signals have the property that $\forall r, n : |s_{rn}|^2 = s_{rn} \cdot s_{rn}^{\star} = c$, with c the *a priori* known squared constant modulus. For multi-modulus signals, we have a more general constraint:

$$\forall r, n : |s_{rn}|^2 = s_{rn} \cdot s_{rn}^{\star} \in \{c_1, \ldots, c_P\},$$

in which P is the number of possible moduli, or, equivalently, the number of concentric circles on which the source samples can lie; e.g., $P = 3$ in the case of rectangular 16-QAM. Hence, one has:

$$\forall r, n : \prod_{p=1}^{P} \left(|s_{rn}|^2 - c_p \right) = 0. \tag{5.3}$$

From these constraints, a minimization problem can be constructed with objective function J:

$$J = \sum_{r=1}^{R} \sum_{n=1}^{N} \left[\prod_{p=1}^{P} \left(|s_{rn}|^2 - c_p \right)^2 \right]. \tag{5.4}$$

Standard optimization techniques can be used to minimize J. Throughout the next subsections, however, we will solve the problem in an algebraic way. The constraint from Eq. (5.3) yields a linear system with structured solutions in subsection 5.3.1. Whereas in [275] an approximate suboptimal quadratic eigenvalue method [358] is used, we will show that by applying a classical subspace-based technique (subsections 5.3.2 and 5.3.3), the problem can be solved by means of coupled CPDs (subsection 5.3.4).

5.3.1 Translation into a constrained set of linear equations

We derive the method for $P = 2$ while the generalization for $P > 2$ is straightforward. Let us consider a single source $\mathbf{s}_r$, drop the subscript r, and substitute (5.2) in (5.3):

$$\forall n: \quad \left(\mathbf{w}^{\mathrm{T}} \mathbf{x}_n \mathbf{x}_n^{\mathrm{H}} \mathbf{w}^{\star} - c_1\right) \left(\mathbf{w}^{\mathrm{T}} \mathbf{x}_n \mathbf{x}_n^{\mathrm{H}} \mathbf{w}^{\star} - c_2\right) = 0, \tag{5.5}$$

then we obtain the following:

$$\forall n : \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n\mathbf{x}_n^{\mathrm{H}}\mathbf{w}^{\star}\right)^2 - (c_1+c_2)\mathbf{w}^{\mathrm{T}}\mathbf{x}_n\mathbf{x}_n^{\mathrm{H}}\mathbf{w}^{\star} + c_1c_2 = 0.$$

As one can show that for arbitrary vectors $\mathbf{d},\mathbf{e} \in \mathbb{C}^I$ and $\mathbf{f},\mathbf{g} \in \mathbb{C}^J$ the equality $(\mathbf{d}^{\mathrm{T}}\mathbf{e})\,(\mathbf{f}^{\mathrm{T}}\mathbf{g}) = (\mathbf{e}\otimes\mathbf{f})^{\mathrm{T}}\,(\mathbf{d}\otimes\mathbf{g})$ holds, we can transform the preceding equation into:

$$\begin{aligned}\forall n : \quad & (\mathbf{x}_n\otimes\mathbf{x}_n^{\star}\otimes\mathbf{x}_n\otimes\mathbf{x}_n^{\star})^{\mathrm{T}}(\mathbf{w}\otimes\mathbf{w}^{\star}\otimes\mathbf{w}\otimes\mathbf{w}^{\star}) - \\ & (c_1+c_2)(\mathbf{x}_n\otimes\mathbf{x}_n^{\star})^{\mathrm{T}}(\mathbf{w}\otimes\mathbf{w}^{\star}) = -c_1c_2.\end{aligned}$$

Stacking for n from 1 to N yields:

$$\begin{bmatrix}(\mathbf{x}_1\otimes\mathbf{x}_1^{\star}\otimes\mathbf{x}_1\otimes\mathbf{x}_1^{\star})^{\mathrm{T}}\\ \vdots \\ (\mathbf{x}_N\otimes\mathbf{x}_N^{\star}\otimes\mathbf{x}_N\otimes\mathbf{x}_N^{\star})^{\mathrm{T}}\end{bmatrix}(\mathbf{w}\otimes\mathbf{w}^{\star}\otimes\mathbf{w}\otimes\mathbf{w}^{\star})$$

$$-(c_1+c_2)\begin{bmatrix}(\mathbf{x}_1\otimes\mathbf{x}_1^{\star})^{\mathrm{T}}\\ \vdots \\ (\mathbf{x}_N\otimes\mathbf{x}_N^{\star})^{\mathrm{T}}\end{bmatrix}(\mathbf{w}\otimes\mathbf{w}^{\star}) = -c_1c_2\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}.$$

This set of equations can be written as follows:

$$\mathbf{R}\mathbf{u} + \mathbf{P}\mathbf{v} = \alpha\mathbf{1}, \tag{5.6}$$

with $\mathbf{1}$ the vector with all ones, the scalar $\alpha = -c_1c_2$ and

$$\mathbf{R} = \begin{bmatrix}(\mathbf{x}_1\otimes\mathbf{x}_1^{\star}\otimes\mathbf{x}_1\otimes\mathbf{x}_1^{\star})^{\mathrm{T}}\\ \vdots \\ (\mathbf{x}_N\otimes\mathbf{x}_N^{\star}\otimes\mathbf{x}_N\otimes\mathbf{x}_N^{\star})^{\mathrm{T}}\end{bmatrix} = (\mathbf{X}\odot\mathbf{X}^{\star}\odot\mathbf{X}\odot\mathbf{X}^{\star})^{\mathrm{T}},$$

$$\mathbf{P} = -(c_1+c_2)\begin{bmatrix}(\mathbf{x}_1\otimes\mathbf{x}_1^{\star})^{\mathrm{T}}\\ \vdots \\ (\mathbf{x}_N\otimes\mathbf{x}_N^{\star})^{\mathrm{T}}\end{bmatrix} = -(c_1+c_2)(\mathbf{X}\odot\mathbf{X}^{\star})^{\mathrm{T}},$$

$$\mathbf{u} = \mathbf{w}\otimes\mathbf{w}^{\star}\otimes\mathbf{w}\otimes\mathbf{w}^{\star}, \quad \mathbf{v} = \mathbf{w}\otimes\mathbf{w}^{\star}. \tag{5.7}$$

Equivalently, one can write

$$\mathbf{T}\mathbf{z} = \alpha\mathbf{1},$$

with $\mathbf{T} = \begin{bmatrix}\mathbf{R} & \mathbf{P}\end{bmatrix}$ and $\mathbf{z} = \begin{bmatrix}\mathbf{u}; \mathbf{v}\end{bmatrix}$. Each vector $\mathbf{w}$ extracting a multi-modulus source satisfies constraint (5.5) and gives a solution to (5.6). Vice versa, if we find a solution vector of system (5.6) with $\mathbf{u}$ and $\mathbf{v}$ constrained as in (5.7), $\mathbf{w}$ is a valid separation vector of the MIMO BSS problem. In total, we need to find R different vectors $\mathbf{w}_r$, yielding the separation matrix $\mathbf{W}$.

5.3.2 Omitting identical columns from the linear system

The Kronecker products introduce identical columns in $\mathbf{R}$, which should be removed. We define a reduced matrix $\mathbf{R}^{\oslash}$ including only the distinct columns of $\mathbf{R}$ multiplied by the number of occurrences, cf. [411]. Equivalently, $\mathbf{u}^{\oslash}$ is introduced containing only the unique elements of $\mathbf{u}$. Then holds:

$$\mathbf{Ru} + \mathbf{Pv} = \alpha\mathbf{1} \quad \Leftrightarrow \quad \mathbf{R}^{\oslash}\mathbf{u}^{\oslash} + \mathbf{Pv} = \alpha\mathbf{1}. \tag{5.8}$$

Considering K observed signals, the matrix $\mathbf{R}^{\oslash}$ has size $N \times \frac{1}{4}K^2(K+1)^2$. Note that there is no need to operate on $\mathbf{P}$ and $\mathbf{v}$, unless both $\mathbf{M}$ and $\mathbf{S}$ are real. Alternatively, we can consider the system $\mathbf{T}^{\oslash}\mathbf{z}^{\oslash} = \alpha\mathbf{1}$ with the matrix $\mathbf{T}^{\oslash} = \begin{bmatrix}\mathbf{R}^{\oslash} & \mathbf{P}\end{bmatrix}$ and the vector $\mathbf{z}^{\oslash} = \begin{bmatrix}\mathbf{u}^{\oslash}; \mathbf{v}\end{bmatrix}$.

5.3.3 Solving the system

The solutions of the system are found in the following procedure through the computation of a null space. To obtain a homogeneous linear system of equations, the right-hand side of (5.8) with all elements equal to α is rotated to the first coordinate axis. The first row of the system can then be dropped.

Let $\mathbf{Q}$ be any unitary matrix such that $\mathbf{Q1} = \begin{bmatrix}N^{\frac{1}{2}}; \mathbf{0}\end{bmatrix}$. $\mathbf{Q}$ can correspond to a Householder or discrete Fourier transformation [165]. Now consider the left multiplication of the system with $\mathbf{Q}$ such that

$$\mathbf{QR}^{\oslash}\mathbf{u}^{\oslash} + \mathbf{QPv} = \alpha\mathbf{Q1} \tag{5.9}$$

$$\Leftrightarrow \quad \begin{bmatrix}\tilde{\mathbf{r}}_1^{\oslash} \\ \tilde{\mathbf{R}}^{\oslash}\end{bmatrix}\mathbf{u}^{\oslash} + \begin{bmatrix}\tilde{\mathbf{p}}_1 \\ \tilde{\mathbf{P}}\end{bmatrix}\mathbf{v} = \begin{bmatrix}\alpha N^{\frac{1}{2}} \\ \mathbf{0}\end{bmatrix}.$$

The vectors $\tilde{\mathbf{r}}_1^{\oslash}$ and $\tilde{\mathbf{p}}_1$ denote the first row of $\mathbf{QR}^{\oslash}$ and $\mathbf{QP}$, respectively, while $\tilde{\mathbf{R}}^{\oslash}$ and $\tilde{\mathbf{P}}$ consist of the other $N-1$ rows. By omitting the first equation, the following system is obtained, with $\tilde{\mathbf{R}}^{\oslash} \in \mathbb{C}^{(N-1)\times\frac{1}{4}K^2(K+1)^2}$ and $\tilde{\mathbf{P}} \in \mathbb{C}^{(N-1)\times K^2}$:

$$\tilde{\mathbf{R}}^{\oslash}\mathbf{u}^{\oslash} + \tilde{\mathbf{P}}\mathbf{v} = \mathbf{0} \Leftrightarrow \begin{bmatrix}\tilde{\mathbf{R}}^{\oslash} & \tilde{\mathbf{P}}\end{bmatrix}\begin{bmatrix}\mathbf{u}^{\oslash} \\ \mathbf{v}\end{bmatrix} = \mathbf{0},$$

$$\Leftrightarrow \quad \tilde{\mathbf{T}}^{\oslash}\mathbf{z}^{\oslash} = \mathbf{0}, \tag{5.10}$$

with the matrix $\tilde{\mathbf{T}}^{\oslash} = \begin{bmatrix}\tilde{\mathbf{R}}^{\oslash} & \tilde{\mathbf{P}}\end{bmatrix} \in \mathbb{C}^{(N-1)\times\frac{1}{4}K^2(K+1)^2+K^2}$. Ignoring the structure in $\mathbf{z}^{\oslash}$, we now focus on the null space of the matrix $\tilde{\mathbf{T}}^{\oslash}$.

In the noiseless case, the dimension of the null space is at least R. Indeed, each separation vector $\mathbf{w}_r$ yields a solution $\mathbf{z}_r^{\oslash} = \left[\mathbf{u}_r^{\oslash}; \mathbf{v}_r\right]$ via (5.10), and these R vectors are linearly independent if the R separation vectors $\mathbf{w}_r$ are independent. This is equivalent with $\mathbf{W}$ having full column rank which is a valid working assumption if $K \geq R$.

Let us first consider the case in which the matrix $\tilde{\mathbf{T}}^{\oslash}$ has more rows than columns minus the null space dimension (i.e., $N - 1 \geq \frac{1}{4}K^2(K+1)^2 + K^2 - R$; we refer to Section 5.4 for an alternative procedure if this is not satisfied). The dimension of the null space is at most R unless there are significant (phase) relations between the source signals. This can occur if the (phases of the) samples are not sufficiently random, cf. the derivations and discussions in [352, 363, 384, 396]. Furthermore, we give the following theorem, which we prove in the appendix of this chapter:

Theorem 5.1. *If* $N - 1 \geq \frac{1}{4}K^2(K+1)^2 + K^2 - R$, $\tilde{\mathbf{T}}^{\oslash}$ *has full column rank for generic* $\mathbf{S}$ *in* (5.1).

Summarizing, one can assume that the dimension of the null space of $\tilde{\mathbf{T}}^{\oslash}$ is exactly R, being the number of separation vectors $\mathbf{w}_r$. Consider a basis $\{\mathbf{f}_1^{\oslash}, \ldots, \mathbf{f}_R^{\oslash}\}$ of the null space of $\tilde{\mathbf{T}}^{\oslash}$. If the observed signals are perturbed by noise, one can use the R smallest singular vectors of $\tilde{\mathbf{T}}^{\oslash}$ for $\{\mathbf{f}_1^{\oslash}, \ldots, \mathbf{f}_R^{\oslash}\}$.

5.3.4 Recovery of the separation matrix

The basis vectors can be partitioned as follows, and also expanded again by reintroducing the repeated elements:

$$\mathbf{f}_r^{\oslash} = \begin{bmatrix} \mathbf{f}_r^{\mathbf{u}^{\oslash}} \\ \mathbf{f}_r^{\mathbf{v}} \end{bmatrix} \in \mathbb{C}^{\frac{1}{4}K^2(K+1)^2+K^2} \;\Leftrightarrow\; \mathbf{f}_r = \begin{bmatrix} \mathbf{f}_r^{\mathbf{u}} \\ \mathbf{f}_r^{\mathbf{v}} \end{bmatrix} \in \mathbb{C}^{K^4+K^2}.$$

As $\mathbf{W}$ is assumed to have full column rank, $\{\mathbf{z}_1, \ldots, \mathbf{z}_R\}$ is a linearly independent set. Hence, it is a basis itself, and each basis vector $\mathbf{f}_r$ is a linear combination of the vectors from $\{\mathbf{z}_1, \ldots, \mathbf{z}_R\}$:

$$\forall r: \quad \mathbf{f}_r = \lambda_{r,1}\mathbf{z}_r + \ldots + \lambda_{r,R}\mathbf{z}_r, \tag{5.11}$$

$$\Leftrightarrow \forall r: \quad \begin{bmatrix} \mathbf{f}_r^{\mathbf{u}} \\ \mathbf{f}_r^{\mathbf{v}} \end{bmatrix} = \lambda_{r,1} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{v}_1 \end{bmatrix} + \ldots + \lambda_{r,R} \begin{bmatrix} \mathbf{u}_R \\ \mathbf{v}_R \end{bmatrix}.$$

The coefficients can be collected in a matrix $\mathbf{\Lambda} \in \mathbb{C}^{R \times R}$. Note that this matrix has full rank. Equivalently, we can write:

$$\begin{aligned} \forall r: \quad \mathbf{f}_r^{\mathbf{u}} &= \lambda_{r,1} \left(\mathbf{w}_1 \otimes \mathbf{w}_1^\star \otimes \mathbf{w}_1 \otimes \mathbf{w}_1^\star \right) + \ldots \\ &\quad + \lambda_{r,R} \left(\mathbf{w}_R \otimes \mathbf{w}_R^\star \otimes \mathbf{w}_R \otimes \mathbf{w}_R^\star \right), \\ \forall r: \quad \mathbf{f}_r^{\mathbf{v}} &= \lambda_{r,1} \left(\mathbf{w}_1 \otimes \mathbf{w}_1^\star \right) + \ldots + \lambda_{r,R} \left(\mathbf{w}_R \otimes \mathbf{w}_R^\star \right). \end{aligned}$$

These equations can now be expressed in a tensor format. Let us reshape the vectors $\mathbf{f}_r^{\mathbf{u}} \in \mathbb{C}^{K^4}$ to fourth-order tensors, and stack them along a fifth mode for $r = 1, \ldots, R$ to obtain the tensor $\mathcal{F}^{\mathbf{u}} \in \mathbb{C}^{K \times K \times K \times K \times R}$. Likewise, we construct a third-order tensor $\mathcal{F}^{\mathbf{v}} \in \mathbb{C}^{K \times K \times R}$ by stacking the matricized versions of $\mathbf{f}_r^{\mathbf{v}}$ along a third mode. Both tensors $\mathcal{F}^{\mathbf{u}}$ and $\mathcal{F}^{\mathbf{v}}$ have rank R, and we obtain two CPDs (note that the column-wise matricization of $\mathbf{w} \otimes \mathbf{w}^\star$ is equal to $\mathbf{w}^\star \otimes \mathbf{w}$):

$$\mathcal{F}^{\mathbf{u}} = \sum_{r=1}^{R} \mathbf{w}_r^\star \otimes \mathbf{w}_r \otimes \mathbf{w}_r^\star \otimes \mathbf{w}_r \otimes \boldsymbol{\lambda}_r = [\![\mathbf{W}^\star, \mathbf{W}, \mathbf{W}^\star, \mathbf{W}, \mathbf{\Lambda}]\!],$$

$$\mathcal{F}^{\mathbf{v}} = \sum_{r=1}^{R} \mathbf{w}_r^\star \otimes \mathbf{w}_r \otimes \boldsymbol{\lambda}_r = [\![\mathbf{W}^\star, \mathbf{W}, \mathbf{\Lambda}]\!].$$

with $\boldsymbol{\lambda}_r$ the rth column of $\mathbf{\Lambda}$. Summarizing, the equations show that the solutions of the original system of (5.6) can be partitioned, and, when reshaped into tensors, can be decomposed using a CPD to recover the separation vectors. This is an interesting result, in multiple ways, as we can now exploit knowledge on tensors and multilinear algebra.

5.3.5 Interpretation in a tensor framework

Let us interpret the previous results in a tensor setting. First, a CPD is *essentially unique* under mild conditions. This means that the factor matrices can be recovered up to scaling and permutation. Note that these are the basic ambiguities in BSS. We can apply the uniqueness results already obtained by Harshman: as it is assumed that the matrix $\mathbf{W}$ has full rank and because $\mathbf{\Lambda}$ does not contain proportional columns (as it has full rank), the decomposition of $\mathcal{F}^{\mathbf{v}}$ is unique [183]. Furthermore, as both $\mathbf{W}$ and $\mathbf{\Lambda}$ have full rank, it is shown in [184] that the CPD of $\mathcal{F}^{\mathbf{v}}$ can be computed by means of the eigenvalue decomposition (EVD). An algebraic technique using the generalized EVD (GEVD) can be found in [242] and references therein.

Hence, the CPD of $\mathcal{F}^{\mathbf{v}}$ is essentially unique and decomposing $\mathcal{F}^{\mathbf{v}}$ is enough to recover $\mathbf{W}$. There is no need to decompose or even construct the larger tensor

$\mathcal{F}^{\mathbf{u}}$. Besides deterministic conditions on $\mathbf{W}$ and/or $\mathbf{\Lambda}$, there also exist generic uniqueness conditions, for which we refer the interested reader to [134, 135].

Second, while it might be sufficient to use $\mathcal{F}^{\mathbf{v}}$, a higher accuracy might be obtained by resorting to the decomposition of the larger tensor $\mathcal{F}^{\mathbf{u}}$.

Third, the CPDs are partially symmetric. For example, $\mathbf{W}$ is a factor matrix of $\mathcal{F}^{\mathbf{u}}$ in the second and fourth mode. The factor matrices $\mathbf{W}$ and $\mathbf{W}^{\star}$ are also related to each other. This can be exploited by customized tensor decomposition algorithms.

Fourth, the tensors are coupled through the matrices $\mathbf{W}$ and $\mathbf{\Lambda}$. This structure can be taken into account by using, e.g., the structured data fusion framework implemented in Tensorlab [333, 392], among others. An overview of data fusion models is given in [6]. By considering the coupling, the uniqueness conditions can be relaxed [345]. The accuracy might increase too.

Furthermore, it is possible to reduce the coupled decompositions of $\mathcal{F}^{\mathbf{u}}$ and $\mathcal{F}^{\mathbf{v}}$ to a single tensor decomposition through relaxation. For example, consider the unfolding of the fifth-order tensor $\mathcal{F}^{\mathbf{u}}$ into a third-order tensor $\tilde{\mathcal{F}}^{\mathbf{u}} \in \mathbb{C}^{K^3 \times K \times R}$. The tensor $\tilde{\mathcal{F}}^{\mathbf{u}}$ then admits the CPD $\tilde{\mathcal{F}}^{\mathbf{u}} = [\![\mathbf{Z}, \mathbf{W}^{\star}, \mathbf{\Lambda}]\!]$ with $\mathbf{Z} = \mathbf{W} \otimes \mathbf{W}^{\star} \otimes \mathbf{W} \in \mathbb{C}^{K^3 \times R}$. Now consider the concatenation of the tensors $\tilde{\mathcal{F}}^{\mathbf{u}}$ and $\mathcal{F}^{\mathbf{v}}$ along the first mode, defining the concatenated tensor $\mathcal{G} \in \mathbb{C}^{(K^3+K) \times K \times R}$. This tensor $\mathcal{G}$ admits the CPD $\mathcal{G} = [\![\mathbf{B}, \mathbf{W}^{\star}, \mathbf{\Lambda}]\!]$ with $\mathbf{B} = \left[(\mathbf{W} \otimes \mathbf{W}^{\star} \otimes \mathbf{W})^{\mathrm{T}} \ \mathbf{W}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{(K^3+K) \times R}$. Alternatively, $\mathcal{G}$ can be reshaped to a tensor $\mathcal{H} \in \mathbb{C}^{(K^2+1) \times K^2 \times R}$ which admits the CPD $\mathcal{H} = [\![\mathbf{C}, \mathbf{D}, \mathbf{\Lambda}]\!]$, with the matrices $\mathbf{D} = \mathbf{W} \otimes \mathbf{W}^{\star} \in \mathbb{C}^{K^2 \times R}$ and $\mathbf{C} = \left[\mathbf{D}; \mathbf{1}\right] \in \mathbb{C}^{(K^2+1) \times R}$ where $\mathbf{1}$ is the vector of size $1 \times R$ with all ones. Hence, the coupled tensor decomposition of $\mathcal{F}^{\mathbf{u}}$ and $\mathcal{F}^{\mathbf{v}}$ has been reduced to a single CPD of $\mathcal{G}$ or $\mathcal{H}$. It is also possible to express the CPD of $\mathcal{F}^{\mathbf{u}}$ as the coupled decomposition of third-order tensors. For details on coupled CPDs we refer the reader to [345, 346].

Finally, note that the structured data fusion framework from Tensorlab allows the user to add regularization to the tensor decompositions or additionally impose structure on $\mathbf{W}$, such as sparseness or nonnegativity.

5.3.6 Note concerning rank deficiency of the mixing matrix

So far, a full-rank mixing matrix $\mathbf{M}$ has been considered, together with its corresponding full-rank separation matrix $\mathbf{W}$. It has been discussed under which circumstances the dimension of the null space of $\tilde{\mathbf{T}}^{\oslash}$ corresponds to the

number of different separation vectors that can be found. In this subsection, it is shown that even if $\mathbf{M}$ is rank-deficient, some source signals can be recovered.

Let us illustrate this with an example that involves four multi-modulus source signals and four observed signals. Let us assume that $\mathbf{m}_4 = \alpha\mathbf{m}_2 + \beta\mathbf{m}_3$ with $\alpha \neq 0, \beta \neq 0$. The vectors $\mathbf{m}_1, \mathbf{m}_2$ and $\mathbf{m}_3$ are linearly independent, and $\mathbf{M} \in \mathbb{C}^{4\times 4}$ has rank 3. Then one can write:

$$\mathbf{X} = \begin{bmatrix}\mathbf{m}_1 & \mathbf{m}_2 & \mathbf{m}_3 & \mathbf{m}_4\end{bmatrix} \begin{bmatrix}\mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \\ \mathbf{s}_4\end{bmatrix} = \underbrace{\begin{bmatrix}\mathbf{m}_1 & \mathbf{m}_2 & \mathbf{m}_3\end{bmatrix}}_{\tilde{\mathbf{M}}} \underbrace{\begin{bmatrix}\mathbf{s}_1 \\ \mathbf{s}_2 + \alpha\mathbf{s}_4 \\ \mathbf{s}_3 + \beta\mathbf{s}_4\end{bmatrix}}_{\tilde{\mathbf{S}}}.$$

Note that, in general, a linear combination of multi-modulus signals is not a multi-modulus signal. Hence, we are unable to recover $\mathbf{s}_2 + \alpha\mathbf{s}_4$ and $\mathbf{s}_3 + \beta\mathbf{s}_4$ using the proposed method. The vector $\mathbf{s}_1$ can be recovered, as there exists a column vector $\mathbf{w} \in \mathbb{C}^4$ such that $\mathbf{w}^{\mathrm{T}}\mathbf{X} = \mathbf{w}^{\mathrm{T}}\tilde{\mathbf{M}}\tilde{\mathbf{S}} = \mathbf{s}_1$.

Generally, let us assume that $\mathbf{M} \in \mathbb{C}^{K\times R}$ has rank $U < R$. $\mathbf{X} = \mathbf{M}\mathbf{S}$ can then be expressed as $\mathbf{X} = \tilde{\mathbf{M}}\tilde{\mathbf{S}}$ with full-rank matrices $\tilde{\mathbf{M}} \in \mathbb{C}^{K\times U}$ and $\tilde{\mathbf{S}} \in \mathbb{C}^{U\times N}$. Let us assume that $\mathbf{M}$ has Z column vectors that are, each, linearly independent of the $R-1$ other vectors. Only the source signals corresponding to these mixing vectors can be recovered, as explained in the example. The null space of $\tilde{\mathbf{T}}^{\oslash}$ has dimension Z, and the coupled tensor decompositions can be used to recover the Z corresponding separation vectors. Note that strictly $Z < U$; if Z was equal to the rank U of $\mathbf{M}$, each of the other $R - Z$ column vectors of $\mathbf{M}$ would be linearly dependent on one or more of the Z column vectors, which is a contradiction.

Note that the source signals corresponding to the other mixing vectors are either removed from $\tilde{\mathbf{S}}$ or are replaced with a linear combination of multi-modulus signals (which is not multi-modulus, in general), as illustrated in the previous example. As the multi-modulus property is lost, they can not be recovered.

5.4 Multi-modulus BSS using a rank-1 detection procedure

We ignored the structure of the vectors $\mathbf{z}_r$ in Section 5.3.3 so that (5.10) could be seen as a linear least squares problem. Let us now exploit the rank-1 structure of $\mathbf{z}_r$ to extract the relevant vectors $\mathbf{w}_r$. The technique of detecting Kronecker-structured vectors (or reshaped rank-1 matrices) is explained in

general in the first subsection. In the second subsection, we explain how the rank-1 detection technique can be used in conjunction with the previously proposed multi-modulus BSS procedure. The method allows us to reduce the number of samples needed.

5.4.1 Rank-1 detection procedure

Consider a matrix $\mathbf{E} \in \mathbb{C}^{IJ \times R_{\text{tot}}}$ that can be written as

$$\mathbf{E} = (\mathbf{A} \odot \mathbf{B})\, \mathbf{G}^{\mathrm{T}}, \tag{5.12}$$

with $\mathbf{A} \in \mathbb{C}^{I \times R}$, $\mathbf{B} \in \mathbb{C}^{J \times R}$, and $\mathbf{G} \in \mathbb{C}^{R_{\text{tot}} \times R}$. We assume that $\mathbf{A} \odot \mathbf{B}$ and $\mathbf{G}$ have full column rank [99]. Equation (5.12) is equivalent to a matrix representation of a polyadic decomposition. Given $\mathbf{E}$, consider now the generic problem of recovering $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{G}$ (up to (counter)scaling and permutation). In [99], this problem was solved by searching for the matrix $\mathbf{W} = (\mathbf{G}^{\mathrm{T}})^{\dagger} \in \mathbb{C}^{R_{\text{tot}} \times R}$ such that

$$\mathbf{E}\mathbf{W} = \mathbf{A} \odot \mathbf{B}.$$

The column space of $\mathbf{E}$ can be represented by a basis of only Kronecker-structured vectors. The goal is then to detect these vectorized rank-1 matrices; hence, the 'rank-1 detection' terminology. Note that an additional mild necessary condition on $\mathbf{A}$ and $\mathbf{B}$ was specified in [99].

Now consider a more general $\mathbf{E} \in \mathbb{C}^{IJ \times R_{\text{tot}}}$ of the form

$$\mathbf{E} = (\mathbf{A} \odot \mathbf{B})\, \mathbf{G}^{\mathrm{T}} + \mathbf{C}\mathbf{H}^{\mathrm{T}} = \begin{bmatrix} \mathbf{A} \odot \mathbf{B} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{G} & \mathbf{H} \end{bmatrix}^{\mathrm{T}}, \tag{5.13}$$

with $\mathbf{C} \in \mathbb{C}^{IJ \times R_{\text{add}}}$ and $\mathbf{H} \in \mathbb{C}^{R_{\text{tot}} \times R_{\text{add}}}$. The subscript of R_{add} stands for 'additional'. We assume both $\begin{bmatrix} \mathbf{A} \odot \mathbf{B} & \mathbf{C} \end{bmatrix}$ and $\begin{bmatrix} \mathbf{G} & \mathbf{H} \end{bmatrix}$ have full column rank. The column space of $\mathbf{E}$ cannot be represented by a basis of only Kronecker-structured vectors anymore — also other vectors are needed — and the problem is not equivalent to finding a CPD anymore.

It is however still possible to detect the different Kronecker-structured vectors in the column space of $\mathbf{E}$ constructed from (5.13) up to machine precision. We will use Algorithm 1 given in [99, Algorithm 2.1, steps 3-10]. The algorithm was discussed in [99] for only $R_{\text{add}} = 0$. Perhaps surprisingly, the algorithm is guaranteed to work in the exact case as well for $R_{\text{add}} > 0$, provided R_{tot} is sufficiently small with respect to I, J (see further).

For a detailed explanation of the algorithm, we refer to [99]. Here, we briefly comment on some steps. Consider the mapping $\Phi : (\mathbf{X}, \mathbf{Y}) \in \mathbb{C}^{I \times J} \times \mathbb{C}^{I \times J} \rightarrow$

$\Phi(\mathbf{X},\mathbf{Y}) \in \mathbb{C}^{I\times I\times J\times J}$ with its resulting tensor $\mathcal{P}$ defined by

$$(\mathcal{P})_{ijkl} = (\Phi(\mathbf{X},\mathbf{Y}))_{ijkl} = x_{ij}y_{il} + y_{ik}x_{jl} - x_{il}y_{jk} - y_{il}x_{jk}.$$

Let us review two properties of the mapping Φ. First, it is clear that the mapping Φ is bilinear in $\mathbf{X}$ and $\mathbf{Y}$, i.e., if $\mathbf{X} = \sum_{m=1}^{M}\alpha_m\mathbf{X}_m$ and $\mathbf{Y} = \sum_{n=1}^{N}\beta_n\mathbf{Y}_n$ then

$$\mathcal{P} = \Phi(\mathbf{X},\mathbf{Y}) = \sum_{m=1}^{M}\sum_{n=1}^{N}\alpha_m\beta_n\Phi(\mathbf{X}_m,\mathbf{Y}_n).$$

Second, consider the case where $\mathbf{Y} = \mathbf{X}$. If and only if $\mathbf{X}$ has rank 1, $\mathcal{P} = \Phi(\mathbf{X},\mathbf{X})$ is an all-zero tensor. Hence, the mapping is able to distinguish between matrices with rank 1 and matrices with rank strictly greater than 1.

By applying Φ on reshaped versions of the columns of $\mathbf{E}$ as detailed in Steps 1 and 2, one obtains a matrix $\mathbf{P} \in \mathbb{C}^{I^2J^2\times R_{\text{tot}}^2}$ in Step 3. Under the necessary assumption that the tensors $\mathcal{P}_{u\tilde{u}}$ are linearly independent for $1 \le u < \tilde{u} \le R_{\text{tot}}$, a CPD can be obtained from the null space of $\mathbf{P}$ as follows. If the tensors are not linearly independent, the method does not work.

Rather than the entire null space of $\mathbf{P}$, we consider a specific subspace. Let us define the 'symmetric null space' of $\mathbf{P}$ as the intersection of the null space and the space spanned by vectorized symmetric matrices. If there are R linearly independent Kronecker-structured vectors in the column space of $\mathbf{E}$, then the dimension of the symmetric null space of $\mathbf{P}$ is equal to R. $\mathbf{M}$ in Step 4 contains a basis of this symmetric null space. Through the multilinearity of Φ, it is shown in [99] that a reshaped tensor version of $\mathbf{M}$ admits the CPD $\mathcal{M} = [\![\mathbf{W},\mathbf{W},\Theta]\!]$.

Note that the null space of $\mathbf{P}^{\oslash} \in \mathbb{C}^{I^2J^2\times\frac{1}{2}R_{\text{tot}}(R_{\text{tot}}+1)}$ is equal to the null space of $\mathbf{P}^{\oslash\text{H}}\mathbf{P}^{\oslash} \in \mathbb{C}^{\frac{1}{2}R_{\text{tot}}(R_{\text{tot}}+1)\times\frac{1}{2}R_{\text{tot}}(R_{\text{tot}}+1)}$; calculating the null space of the matrix $\mathbf{P}^{\oslash\text{H}}\mathbf{P}^{\oslash}$ might be computationally more efficient as the latter matrix is smaller. $\mathbf{P}^{\oslash\text{H}}\mathbf{P}^{\oslash}$ can also be constructed in an efficient way. Additionally, the number of rows from $\mathbf{P}^{\oslash}$ can be reduced to $\frac{1}{4}I(I+1)J(J+1)$ by removing redundant rows. Step 3 in Algorithm 1 can be computationally demanding for large R_{tot}, as the number of elements in $\mathbf{P}^{\oslash\text{T}}\mathbf{P}^{\oslash}$ is of the order of magnitude $\mathcal{O}(R_{\text{tot}}^4)$.

A critical assumption in Algorithm 1 is that the tensors $\mathcal{P}_{u\tilde{u}}$ are linearly independent for $u < \tilde{u}$. A generic bound was given in [99, p. 656] for $R_{\text{add}} = 0$:

$$\frac{R(R-1)}{2} \le \frac{I(I-1)J(J-1)}{4}. \tag{5.14}$$

We now conjecture the following bound for $R_{\text{add}} \ge 0$:

$$\frac{R_{\text{tot}}(R_{\text{tot}}+1)}{2} - R \le \frac{I(I-1)J(J-1)}{4}. \tag{5.15}$$

We have verified this bound empirically for various combinations of R, R_{add}, I and J. One can see that the bound in (5.14) is the specific case for $R_{\text{add}} = 0$ or, equivalently, $R_{\text{tot}} = R$.

Algorithm 1: Rank-1 detection procedure

Input : Matrix $\mathbf{E}$

Output: Matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{W}$ such that $\mathbf{EW} = \mathbf{A} \odot \mathbf{B}$

1. Construct a tensor $\mathcal{E} \in \mathbb{C}^{I \times J \times R_{\text{tot}}}$ as follows:

$$\forall i, j, u : \quad (\mathcal{E})_{iju} = (\mathbf{E})_{(i-1)J+j,u} \,.$$

2. Compute $\mathcal{P}_{u\tilde{u}} \in \mathbb{C}^{I \times I \times J \times J}, 1 \leq u, \tilde{u} \leq R_{\text{tot}}$ as follows:

$$\forall i, j, k, l : \quad (\mathcal{P}_{u\tilde{u}})_{ijkl} = e_{iku}e_{jl\tilde{u}} + e_{ik\tilde{u}}e_{jlu} - e_{ilu}e_{jk\tilde{u}} - e_{il\tilde{u}}e_{jku}.$$

3. Reshape each tensor $\mathcal{P}_{u\tilde{u}}$ to a vector of length I^2J^2 and stack the different vectors in the columns of a matrix $\mathbf{P} \in \mathbb{C}^{I^2J^2 \times R_{\text{tot}}^2}$. Compute the symmetric null space of $\mathbf{P}$ by
   - Removing identical columns to obtain $\mathbf{P}^{\oslash}$;
   - Computing the null space of $\mathbf{P}^{\oslash}$ (or equivalently $\mathbf{P}^{\oslash \text{H}}\mathbf{P}^{\oslash}$);
   - Expanding the R computed null space vectors again as discussed in subsection 5.3.4 to obtain a matrix $\mathbf{M} \in \mathbb{C}^{R_{\text{tot}}^2 \times R}$.
4. Reshape the matrix $\mathbf{M}$ to the tensor $\mathcal{M} \in \mathbb{C}^{R_{\text{tot}} \times R_{\text{tot}} \times R}$.
5. Compute a rank-R CPD $\mathcal{M} = [\![\mathbf{W}, \mathbf{W}, \mathbf{\Theta}]\!]$.
6. Compute $\mathbf{Z} = \mathbf{EW}$. Reshape each vector $\mathbf{z}_r$ to a matrix of size $J \times I$ and compute a best rank-1 approximation to obtain $\mathbf{b}_r$ and $\mathbf{a}_r$.

5.4.2 Application in the multi-modulus setting

Recall the problem in subsection 5.3.4 of finding R structured vectors $\mathbf{z}_r = [\mathbf{w}_r \otimes \mathbf{w}_r^\star \otimes \mathbf{w}_r \otimes \mathbf{w}_r^\star;\, \mathbf{w}_r \otimes \mathbf{w}_r^\star]$ in the null space of $\mathbf{P}$. One can see that $\mathbf{z}_r$ is a Kronecker-structured vector, as we can write $\mathbf{z}_r = [\mathbf{w}_r \otimes \mathbf{w}_r^\star;\, 1] \otimes (\mathbf{w}_r \otimes \mathbf{w}_r^\star) = [\mathbf{v}_r;\, 1] \otimes \mathbf{v}_r$.

In subsection 5.3.3 a tall $\tilde{\mathbf{T}}^{\oslash}$ is assumed, i.e., $N - 1 \geq \frac{1}{4}K^2(K+1)^2 + K^2 - R$. This limits the dimension of the null space of $\tilde{\mathbf{T}}^{\oslash}$ to R such that the null space is

exactly spanned by the vectors $\mathbf{z}_r$. Using $\mathbf{z}_r = [\mathbf{v}_r;\ 1] \otimes \mathbf{v}_r$, the system in (5.11) can be expressed in the form (5.12). This can be solved using the algebraic Algorithm 1 as discussed in [99]. As $\mathbf{a}_r = [\mathbf{w}_r \otimes \mathbf{w}_r^\star;\ 1]$ and $\mathbf{b}_r = \mathbf{w}_r \otimes \mathbf{w}_r^\star$, the different $\mathbf{w}_r$ can be recovered up to the standard scaling and permutation ambiguities by using a best rank-1 approximation on a reshaped version of either $\mathbf{a}_r$ or $\mathbf{b}_r$, or in a coupled way.

However, if $N - 1 < \frac{1}{4}K^2(K+1)^2 + K^2 - R$, the dimension of the null space is strictly larger than R, simply because $\tilde{\mathbf{T}}^{\oslash}$ is a wide matrix. The null space is not only spanned by the vectors $\mathbf{z}_r$ but also by other arbitrary vectors. Let us define the dimension as $R_{\text{tot}} > R$. Generically, $R_{\text{tot}} = R + R_{\text{add}}$ with $R_{\text{add}} = \left(\frac{1}{4}K^2(K+1)^2 + K^2 - R\right) - (N-1)$. Rather than (5.12), (5.13) now describes the structure of the null space of $\tilde{\mathbf{T}}^{\oslash}$. While the system in (5.11) and thus the approach in Section 5.3 is not valid in this setting, Algorithm 1 can still be used as explained in Section 5.4.1 to recover the R Kronecker-structured vectors $\mathbf{z}_r = [\mathbf{v}_r;\ 1] \otimes \mathbf{v}_r$ from the null space of $\tilde{\mathbf{T}}^{\oslash}$ with dimension $R_{\text{tot}} > R$.

The general bound in (5.15) does not apply directly because the matrices $\mathbf{A}$ and $\mathbf{B}$ in (5.13) have a particular structure, as explained previously. We conjecture the following bound in the multi-modulus context, which we have verified for different values of R and N:

$$\frac{R_{\text{tot}}(R_{\text{tot}}+1)}{2} - R \leq \frac{K^2(K^2-1)}{4}\left[\frac{K^2(K^2-1)}{2} + 1\right] - K\binom{K}{4},$$

with $R_{\text{tot}} = R + R_{\text{add}} = \frac{1}{4}K^2(K+1)^2 + K^2 - N + 1$ and with $\binom{K}{4} = \frac{K!}{4!(K-4)!}$ if $R \geq 4$ and zero otherwise. The required number of samples for 2, 3 and 4 source signals, each with two possible source moduli, is reduced to 8, 11 and 20 samples instead of 12, 43 and 113 samples, respectively.

The mapping was applied by considering the Kronecker structure $\mathbf{z}_r = [\mathbf{v}_r; 1] \otimes \mathbf{v}_r$. To obtain a more powerful version of the rank-1 detection technique, one can take into account the full Kronecker structure in $\mathbf{z}_r = [\mathbf{w}_r \otimes \mathbf{w}_r^\star \otimes \mathbf{w}_r \otimes \mathbf{w}_r^\star;\ \mathbf{w}_r \otimes \mathbf{w}_r^\star]$, working in analogy with [345, 346]. The mapping Φ is applied four times in total: three times on the first part of $\mathbf{z}_r$, focusing on $(\mathbf{w}_r \otimes \mathbf{w}_r^\star \otimes \mathbf{w}_r) \otimes \mathbf{w}_r^\star$, $(\mathbf{w}_r \otimes \mathbf{w}_r^\star) \otimes (\mathbf{w}_r \otimes \mathbf{w}_r^\star)$ and $\mathbf{w}_r \otimes (\mathbf{w}_r^\star \otimes \mathbf{w}_r \otimes \mathbf{w}_r^\star)$, and once on the second part of $\mathbf{z}_r$ consisting of $\mathbf{w}_r \otimes \mathbf{w}_r^\star$. Each mapping will provide a different matrix $\mathbf{P}$ in Step 3. By vertically stacking these matrices in a large matrix, the null space of interest can be estimated. A more relaxed bound can be obtained on the required number of samples. A detailed derivation is considered to be out of scope of this chapter.

Summarizing, we have shown that the rank-1 detection procedure can extract the separation vectors from the null space of $\tilde{\mathbf{T}}^{\oslash}$, even if the null space is spanned by not only Kronecker-structured vectors, enabling a reduction of the

number of samples required for the separation of the multi-modulus source signals.

5.5 Blind deconvolution of multi-modulus signals

In this section, we consider the equalization of MIMO systems of multi-modulus signals. The problem is more general than the instantaneous problem from Section 5.3. The data model is first discussed in Subsection 5.5.1. Subsections 5.5.2 and 5.5.3 discuss two different methods to solve the BSI problem.

5.5.1 Data model

We consider a following data model with system order L_s:

$$\forall k, n: \; x_k(n) = \sum_{r=1}^{R} \sum_{l=0}^{L_s} h_{kr}^{(l)} s_r(n-l).$$

where the noise term is omitted. The model can be written as

$$\forall n: \; \mathbf{x}_n = \sum_{l=0}^{L_s} \mathbf{H}^{(l)} \mathbf{s}_{n-l},$$

in which the matrices $\mathbf{H}^{(l)} \in \mathbb{C}^{K \times R}$ contain the unknown system coefficients for $l = 0, \ldots, L_s$; hence, the Z-transform $\mathcal{H}$ of the system can be written as follows:

$$\mathbf{H}[z] = \mathbf{H}^{(0)} + \mathbf{H}^{(1)} z^{-1} + \ldots + \mathbf{H}^{(L_s)} z^{-L_s}.$$

For $L_s = 0$, one obtains the instantaneous case from Section 5.3. Alternatively, consider $\mathbf{H} = \begin{bmatrix} \mathbf{H}^{(0)} & \cdots & \mathbf{H}^{(L_s)} \end{bmatrix} \in \mathbb{C}^{K \times (L_s+1)R}$ and let us stack the vectors $\mathbf{s}_n$, $\mathbf{s}_{n-1}$, $\ldots$, $\mathbf{s}_{n-L_s}$ in one large vector $\mathbf{s}_n^{\square} \in \mathbb{C}^{(L_s+1)R}$ for $n = L_s + 1, \ldots, N$. By concatenating the vectors $\mathbf{s}_n^{\square}$, a block-Toeplitz matrix $\mathbf{S}^{\square} \in \mathbb{C}^{(L_s+1)R \times (N-L_s)}$ is obtained such that

$$\mathbf{X} = \mathbf{H}\mathbf{S}^{\square}. \tag{5.16}$$

The superscript $\cdot^{\square}$ stands for the block-Toeplitz encapsulation.

We assume that it is possible to equalize the channel by means of a finite impulse response (FIR) filter $\mathbf{F}(z)$ with filter order L_x:

$$\mathbf{s}_n = \sum_{l=0}^{L_x} \mathbf{F}^{(l)} \mathbf{x}_{n-l},$$

with $\mathbf{F}^{(l)} \in \mathbb{C}^{R \times K}$ for $l = 0, \ldots, L_x$. This assumption is valid when $K > R$ and under some additional conditions on the coefficients [254, 386]. This bound can be further relaxed to $K \geq R$ when considering paraunitary systems as these can be equalized by an FIR filter of the same length; hence, $L_s = L_x = L$ [100, 371]. We consider general non-paraunitary systems in this section unless stated otherwise. Again, by collecting the vectors $\mathbf{x}_n$, $\mathbf{x}_{n-1}$, $\ldots$, $\mathbf{x}_{n-L_x}$ in a block-Toeplitz matrix $\mathbf{X}^{\square} \in \mathbb{C}^{(L_x+1)K \times (N-L_x)}$, it is possible to write $\mathbf{S} = \mathbf{F}\mathbf{X}^{\square}$ with $\mathbf{F} = \begin{bmatrix} \mathbf{F}^{(0)} & \cdots & \mathbf{F}^{(L_x)} \end{bmatrix} \in \mathbb{C}^{R \times (L_x+1)K}$.

5.5.2 Deconvolution using an increased number of source signals

Consider the case where $K \geq R(L_s + 1)$; the matrix $\mathbf{H}$ has then more rows than columns and one can write $\mathbf{S}^{\square} = \mathbf{W}\mathbf{X}$ with $\mathbf{W} = \mathbf{H}^{\dagger} \in \mathbb{C}^{(L_s+1)R \times K}$. As a delayed multi-modulus signal is still multi-modulus, the techniques from Section 5.3 and 5.4 can be used, assuming there are $(L_s + 1)R$ different source signals instead of only R signals. Note that the block-Toeplitz structure in $\mathbf{S}^{\square}$ is not taken into account.

Otherwise, if $R < K < (L_s + 1)R$, a smoothing technique can be applied in a preprocessing step. By delaying the observed signals, one can artificially construct a system such that a tall mixing matrix $\tilde{\mathbf{H}}$ is obtained. The techniques from Section 5.3 and 5.4 can again be used to recover the original source signals. For further details on the smoothing technique we refer to e.g. [254, 386].

5.5.3 Deconvolution using the original number of source signals by exploiting the block-Toeplitz structure

The number of samples required for a unique decomposition depends directly on the number of source signals R, as discussed in Sections 5.3 and 5.4. By exploiting the block-Toeplitz structure of $\mathbf{S}^{\square}$, it is possible to convert the deconvolution problem into an instantaneous BSS problem involving only R source signals instead of $(L_s + 1)R$, as required in the previous section.

Consider the problem from (5.16). Ignoring the multi-modulus constraint, this is a block-Toeplitz factorization of the matrix $\mathbf{X}$. This is just one factorization; let us consider an alternative factorization of $\mathbf{X}$:

$$\mathbf{X} = \mathbf{H}\mathbf{S}^{\square} = \mathbf{H}\mathbf{Z}^{-1}\mathbf{Z}\mathbf{S}^{\square} = \mathbf{U}\mathbf{V}^{\square},$$

with $\mathbf{U} = \mathbf{H}\mathbf{Z}^{-1}$, $\mathbf{V}^{\square} = \mathbf{Z}\mathbf{S}^{\square}$ and $\mathbf{Z} \in \mathbb{C}^{(L_s+1)R \times (L_s+1)R}$ an arbitrary invertible matrix. Consider the following matrix $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{s}_{L_s+2} & \cdots & \cdots & \mathbf{s}_{N-L_s-1} & \cdots & \mathbf{s}_N \\ \vdots & \ddots & & & \ddots & \vdots \\ \mathbf{s}_1 & \cdots & \mathbf{s}_{L_s+2} & \cdots & \cdots & \mathbf{s}_{N-L_s-1} \end{bmatrix}^{\mathrm{T}}.$$

We can now proceed with a result that can for instance be found in [253, 254, 338]: if both $\mathbf{H}$ and $\mathbf{\Sigma}$ have full column rank, then $\mathbf{X} = \mathbf{U}\mathbf{V}^{\square}$ is a block-Toeplitz factorization of $\mathbf{X}$ if and only if $\mathbf{Z}$ is of the following form:

$$\mathbf{Z} = \mathbf{I}_{L_s+1} \otimes \mathbf{G},$$

in which $\mathbf{G} \in \mathbb{C}^{R \times R}$ is an invertible matrix. Let $\mathbf{V}$ be the data matrix corresponding to the block-Toeplitz matrix $\mathbf{V}^{\square}$, in the same way as $\mathbf{S}^{\square}$ is constructed from $\mathbf{S}$. Then as $\mathbf{V}^{\square} = (\mathbf{I}_{L_s+1} \otimes \mathbf{G})\,\mathbf{S}^{\square}$, we have that $\mathbf{V} = \mathbf{G}\mathbf{S}$. Hence, $\mathbf{V}^{\square}$ can be found from a block-Toeplitz factorization of $\mathbf{X}$ (which is a linear least squares problem) and $\mathbf{V}$ can be derived from $\mathbf{V}^{\square}$. After $\mathbf{V}$ has been determined, what remains to do is to find an invertible matrix $\mathbf{G}$ such that $\mathbf{S}$ is multi-modulus. This is exactly the instantaneous BSS problem which has been discussed in Sections 5.3 and 5.4.

Stated otherwise: if $\mathbf{H}$ and $\mathbf{\Sigma}$ have full column rank, then the block-Toeplitz matrix factorization $\mathbf{X} = \mathbf{U}\mathbf{V}^{\square}$ reduces the deconvolution to an instantaneous BSS problem that involves only R source signals. For the solution of the linear least squares block-Toeplitz problem to be unique, it is necessary that $\mathbf{\Sigma}$ has full column rank, which we can interpret as a working assumption. Second, the matrix $\mathbf{H} \in \mathbb{C}^{K \times (L_s+1)R}$ should have full column rank which requires that $K \geq (L_s+1)R$. Otherwise, if $R < K < (L_s+1)R$, it is again possible to apply the smoothing technique from Section 5.5.2.

5.6 Simulations

We investigate the behavior of the proposed multi-modulus method for a varying number of samples and signal-to-noise ratio (SNR), as well as the behavior of Algorithm 1. The multi-modulus technique is compared to three other methods:

robustICA [412], ACMA [384] and a recently proposed MMA technique using Givens rotations [316, 317]. ICA has already been used before in a constant or multi-modulus context [279], and its use is valid for the following simulations as the source signals are mutually stochastically independent.

To calculate a CPD, various approaches exist such as the popular alternating least squares (ALS) algorithm. We use a nonlinear least squares (NLS) approach, implemented in Tensorlab [392]. A generalized eigenvalue decomposition is used for the initialization, together with five random initializations. Each time, the final solution with the minimal cost function value is retained. The observed signals are first prewhitened.

For the coupled decompositions, the coupling constraints are incorporated using the dedicated `ccpd_nls` algorithm from Tensorlab, with the following cost function:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \frac{\omega_1}{2} ||\mathcal{F}^{\mathbf{u}} - [\![\mathbf{B},\mathbf{A},\mathbf{B},\mathbf{A},\mathbf{C}]\!]||^2 + \frac{\omega_2}{2} ||\mathcal{F}^{\mathbf{v}} - [\![\mathbf{B},\mathbf{A},\mathbf{C}]\!]||^2 ,$$

in which the weights ω_1 and ω_2 are the inverse squared Frobenius norms of $\mathcal{F}^{\mathbf{u}}$ and $\mathcal{F}^{\mathbf{v}}$, respectively. The complex conjugation symmetry is not exploited; hence, the matrices $\mathbf{W}$ and $\mathbf{W}^\star$ are considered as different factor matrices $\mathbf{A}$ and $\mathbf{B}$, respectively. The matrix $\mathbf{\Lambda}$ is estimated as $\mathbf{C}$. For the initialization, we use a combination of the GEVD initializations of the lower-order and higher-order tensor, together with five additional random initializations.

To determine the relative error, we correct for scaling and permutation (the default indeterminacies in BSS) with respect to the theoretical sources. The relative error is then defined as the relative difference in Frobenius norm, e.g., if $\hat{\mathbf{S}}$ are the recovered sources after this step, we have a relative error $\epsilon_{\mathbf{S}} = \|\mathbf{S} - \hat{\mathbf{S}}\|/\|\mathbf{S}\|$. Second, the SNR is defined as the ratio of the power of the signal to the power of the Gaussian additive noise. For each experiment, the medians of the relative errors are given across 100 simulations. In each run, new realizations of source signals, noise signals and mixing matrix / channel coefficients are generated.

5.6.1 Simulation for rank-1 detection algorithm

A first simulation considers the rank-1 detection method in Algorithm 1. The matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{G}$ and $\mathbf{H}$ from (5.13) are randomly generated using a Gaussian distribution with $I = J = 10$, $U = 40$ and $R = 2$. Hence, while the dimension of the column space of $\mathbf{E}$ from (5.13) is 40, the dimension of the intersection of the column space and the space of Kronecker-structured vectors is only 2. Algorithm 1 is compared with a method which extracts the

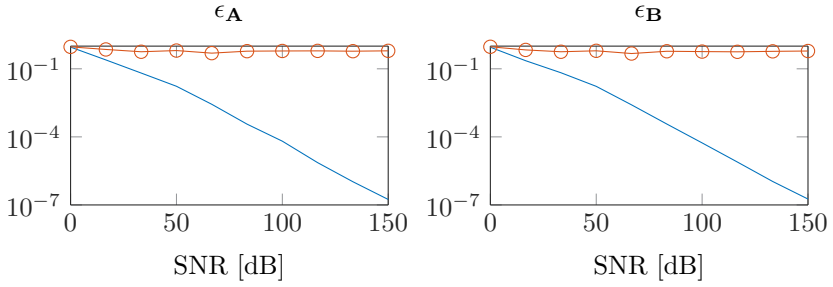


Figure 5.2: Relative errors on $\mathbf{A}$ and $\mathbf{B}$ for the rank-1 detection simulation from Section 5.6.1 using Algorithm 1 (——) and using a CPD with rank $R = 2$ (—⊖—) in function of the SNR.

two Kronecker-structured vectors by applying a CPD with rank $R = 2$ on a reshaped version of $\mathbf{E}$. Fig. 5.2 illustrates the estimation error on $\mathbf{A}$ and $\mathbf{B}$ in function of the SNR on $\mathbf{E}$. It shows that the CPD approach fails as the influence of $\mathbf{C}\mathbf{H}^{\mathrm{T}}$ is not negligible, while Algorithm 1 delivers good results.

5.6.2 Simulations for the instantaneous case

A first experiment regarding multi-modulus BSS considers two ($R = 2$) multi-modulus continuous-phase shift keying (CPSK) source signals of 100 samples each ($N = 100$), mixed into two ($K = 2$) observed signals. Each source sample has a uniform random phase drawn from $[0, 2\pi[$ and a modulus of 1 ($c_1 = 1$) or 2 ($c_2 = 4$) with equal probability. A unitary mixing matrix is used. Fig. 5.3 visualizes the relative errors on the mixture matrix and the source signals for varying SNRs. The proposed technique shows optimal asymptotic performance for increasing SNR, in contrast to the other methods. The proposed technique performs slightly worse at low SNR in terms of mixing matrix error; however, more important is the source recovery error, where the difference is negligible. The accuracy of ICA is limited because of the statistics estimation error. CMA is not suited in a multi-modulus context, and MMA optimizes a non-suitable cost function.

In a second experiment, we take a similar setting but vary the number of samples for an SNR of 20 dB and 30 dB, as shown in Fig. 5.4. The relative error is only weakly dependent on the number of samples and the solutions are about as good as the zero-forcing solutions. The experiment shows that not many samples are needed to reach asymptotic performance.

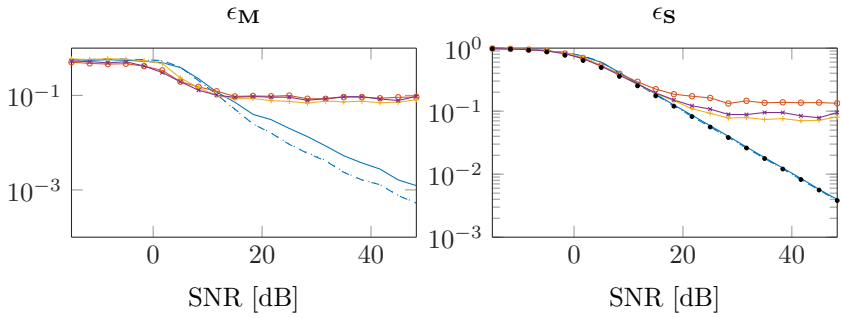


Figure 5.3: Results for the first experiment in Section 5.6.2 with CPSK source signals. Both the relative error on the mixing matrix (left) and on the source matrix (right) are shown for varying SNRs. The results obtained by the proposed methods are shown in blue. We discuss variants using only the lower-order tensor (——), only the higher-order tensor (- - -) and both in a coupled way (·····). The coupled and higher-order methods show a similar performance. They are compared with ICA (—○—), ACMA (—+—) and MMA using Givens rotations (—×—). The black dots (•) represent the case in which the exact mixing matrix is used to recover the source signals, a.k.a. the zero-forcing solution.

In the third experiment, we consider four rectangular 16-QAM source signals with $N = 2000$ samples. The samples are drawn from $\pm\{1, 3\} \pm \{1, 3\}i$ with equal probability, having a squared modulus of $c_1 = 2$, $c_2 = 10$ or $c_3 = 18$; hence, $P = 3$. We use a generalization of the technique for two moduli explained in Section 5.3, allowing now three different moduli. Four instantaneously mixed signals are observed. For this experiment, a mixing matrix with condition number 10 is used to illustrate the performance in ill-conditioned situations. For $P = 3$, a set of coupled decompositions of a third-order tensor, a fifth-order tensor and a seventh-order tensor is obtained. Fig. 5.5 visualizes the results when decomposing the third-order tensor, the seventh-order tensor and both tensors coupled. In terms of the source error, the solutions are about as good as the zero-forcing solutions for increasing SNR and are only slightly worse for low SNRs. Fig. 5.6 visualizes the source signals, the observed signals and the recovered signals at 40 dB.

5.6.3 Simulations for the convolutive case

For the convolutive case, we consider a single experiment with two ($R = 3$) multi-modulus CPSK source signals of 500 samples each. The samples have a

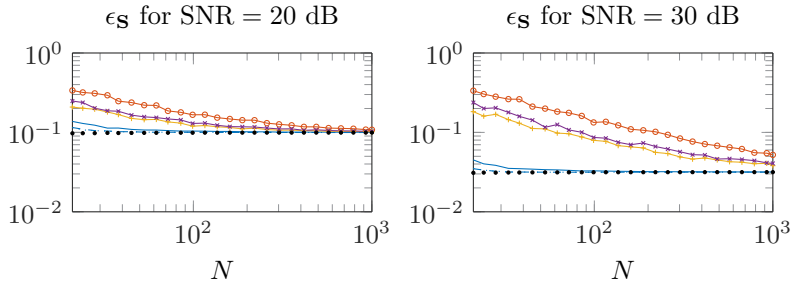


Figure 5.4: Results for the second experiment in Section 5.6.2. The relative error on the source matrix is shown for SNRs of 20 dB (left) and 30 dB (right) for an increasing number of samples N. The labels are the same as in Fig. 5.3.

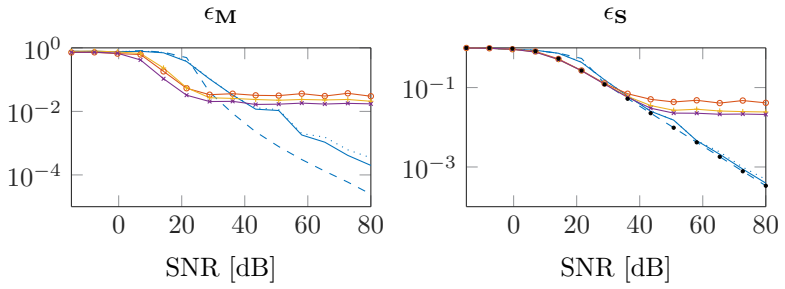


Figure 5.5: Results for the third experiment in Section 5.6.2 with 16-QAM source signals. Both the relative error on the mixing matrix (left) and on the source matrix (right) are shown for varying SNR. The labels are the same as in Fig. 5.3.

squared modulus of $c_1 = 1$ or $c_2 = 4$ and a random phase. A random system is used of order $L_s = 2$. Six signals ($K = 6$) are observed.

As it is not straightforward to compare transfer functions, we only report the error on the recovered deconvolved source signals in Fig. 5.7. Both the results from the method of subsection 5.5.2 with $(L_s + 1)R$ and the block-Toeplitz factorization method of subsection 5.5.3 with R source signals are given. For the former method, the technique works well for middle to high SNR, and performs slightly worse compared to the other algorithms for low SNR. The block-Toeplitz factorization procedure performs optimally.

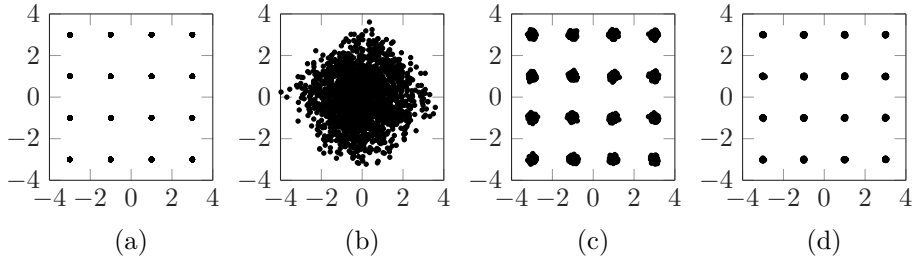


Figure 5.6: Visualization in the complex plane of the different signals in the third experiment from Section 5.6.2 at 40 dB. In (a), one of the 16-QAM source signals is shown. One of the observed mixed signals is shown in (b). In (c) and (d), an estimate of a source signal is shown (after compensating for scaling and permutation) by applying the ICA technique and the proposed technique, respectively.

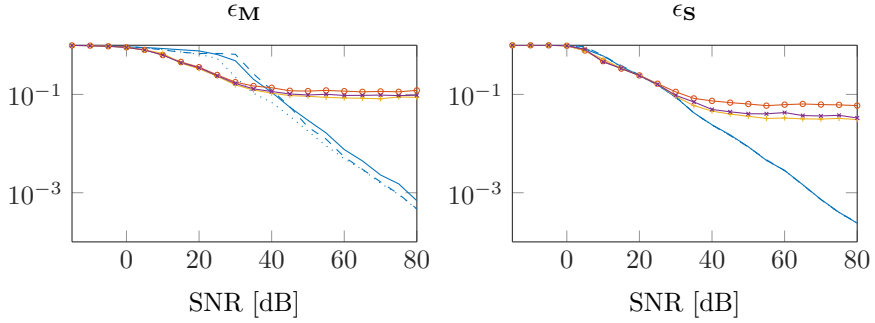


Figure 5.7: Results for the experiment from Section 5.6.3 with the blind deconvolution of two multi-modulus signals. On the left, the procedure of Section 5.5.2 is applied with $(L_s + 1)R$ source signals. On the right, the procedure of Section 5.5.3 based on a block-Toeplitz factorization is applied with R source signals. The latter method clearly outperforms the former. The labels are the same as in Fig. 5.3.

5.7 Conclusion

We have proposed a new technique for multiple-input-multiple-output blind signal separation and blind system identification of multi-modulus signals such as 16-QAM signals. The method includes an analytical transformation to a set of CPDs by using subspace methods. We have shown that a number of source signals can still be recovered in the case of a rank-deficient mixing matrix. As a side-result, an existing rank-1 detection procedure has been generalized to find Kronecker-structured vectors in a space spanned by both Kronecker-structured vectors and arbitrary vectors, and we include a conjectured bound on the number of vectors that can be handled. The generalized technique has been applied in the multi-modulus context to allow for a reduction in number of samples required for separation. The proposed methods can be interpreted as generalizations of the analytical constant modulus algorithm (ACMA). Some advantages of the tensor framework have been discussed, and the algorithm has been tested in different situations and compared to methods from the literature. An algebraic or exact solution can be obtained in the noiseless case for the proposed method, and the method reaches optimal asymptotic performance for increasing SNR.

In this chapter, each source signal was assumed to be multi-modulus. If only Z source signals with $Z < R$ are multi-modulus, the dimension of the null space of $\tilde{\mathbf{T}}^{\oslash}$ in (5.10) is Z in general. The Z corresponding separation vectors can then be found using the techniques proposed in this chapter, despite the presence of non-multi-modulus source signals.

In [385], a method was suggested for a BSS problem in which the source signal samples are assumed to have either constant modulus or zero modulus. The technique proposed in this chapter can be extended to allow such signals. Eq. (5.5) then changes to

$$\forall n: \quad (\mathbf{w}^{\mathrm{T}}\mathbf{x}_n\mathbf{x}_n^{\mathrm{H}}\mathbf{w}^{\star} - c_1)\,(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n\mathbf{x}_n^{\mathrm{H}}\mathbf{w}^{\star} - c_2)\,\mathbf{w}^{\mathrm{T}}\mathbf{x}_n\mathbf{x}_n^{\mathrm{H}}\mathbf{w}^{\star} = 0.$$

One can work in analogy with Section 5.3 to obtain an algebraic method.

Comparison with the MMA cost function

In [282], a cost function with multiple dispersion constants D_R and D_I (also defined as the in-phase and quadrature moduli, respectively) was designed for

separating QAM signals:

$$J = \sum_{r=1}^{R} \sum_{n=1}^{N} \left[\left(\Re\{s_{rn}\}^2 - D_R^2 \right)^2 + \left(\Im\{s_{rn}\}^2 - D_I^2 \right)^2 \right]. \tag{5.17}$$

It has been coined the multi-modulus algorithm (MMA) cost function in [401, 402]. Typically, one uses $D_R = D_I$, as a compromise value which acts to project the constellation points onto the same circle. The cost function has been reused in many subsequent publications on the separation or deconvolution of multi-modulus signals such as [98, 187, 198].

Let us recall the cost function from Eq. (5.4) for $P = 2$:

$$\tilde{J} = \sum_{r=1}^{R} \sum_{n=1}^{N} \left[\left(|s_{rn}|^2 - c_1 \right) \left(|s_{rn}|^2 - c_2 \right) \right]. \tag{5.18}$$

This cost function is only implicitly used in this paper. Given that $|s_{rn}|^2 = \Re\{s_{rn}\}^2 + \Im\{s_{rn}\}^2$, one can rework Eq. (5.18) to the following:

$$r\tilde{J} = \sum_{r=1}^{R} \sum_{n=1}^{N} \left[\left(\Re\{s_{rn}\}^2 - F_R^2 \right)^2 + \left(\Im\{s_{rn}\}^2 - F_I^2 \right)^2 \right.$$
$$\left. + \frac{1}{2} \left(4\Re\{s_{rn}\}^2 \Im\{s_{rn}\}^2 - c_1^2 - c_2^2 \right) \right]$$

with $F_R = F_I = \sqrt{\frac{c_1 + c_2}{2}}$. The constant terms c_1^2 and c_2^2 are not relevant for the minimization; hence, they can be omitted. We have now shown that both cost functions are not equivalent, but rather differ in one term:

$$\tilde{J} \equiv J + 2 \sum_{r=1}^{R} \sum_{n=1}^{N} \Re\{s_{rn}\}^2 \Im\{s_{rn}\}^2.$$

The results in the paper have shown that the cost function J in (5.17) is not able to find the optimal solution, while the proposed cost function $\tilde{J}$ in (5.18) allows optimal asymptotic performance.

Proof of Theorem 5.1

We first give the following lemma from [135, 174, 207, 344] which is used in the proof:

Lemma 5.1. *Given an analytic function $f : \mathbb{C}^N \to \mathbb{C}$. If there exists an element $\mathbf{x} \in \mathbb{C}^N$ such that $f(\mathbf{x}) \neq 0$, then the set $\{\mathbf{x}|f(\mathbf{x}) = 0\}$ is of Lebesgue measure zero.*

Let us also define the following matrices related to (5.6):

$$\mathbf{Z}_{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \odot \mathbf{X}^\star \odot \mathbf{X} \odot \mathbf{X}^\star \\ \mathbf{X} \odot \mathbf{X}^\star \end{bmatrix}, \qquad \mathbf{Z}_{\mathbf{S}} = \begin{bmatrix} \mathbf{S} \odot \mathbf{S}^\star \odot \mathbf{S} \odot \mathbf{S}^\star \\ \mathbf{S} \odot \mathbf{S}^\star \end{bmatrix},$$

$$\mathbf{Z}_{\mathbf{X}}^{\boxslash} = \begin{bmatrix} (\mathbf{X} \odot \mathbf{X}^\star \odot \mathbf{X} \odot \mathbf{X}^\star)^{\boxslash} \\ \mathbf{X} \odot \mathbf{X}^\star \end{bmatrix}, \quad \mathbf{Z}_{\mathbf{S}}^{\boxslash} = \begin{bmatrix} (\mathbf{S} \odot \mathbf{S}^\star \odot \mathbf{S} \odot \mathbf{S}^\star)^{\boxslash} \\ \mathbf{S} \odot \mathbf{S}^\star \end{bmatrix}.$$

It can be seen that $\mathbf{Z}_{\mathbf{X}}^{\boxslash} = [\mathbf{R}^{\oslash}\ \mathbf{P}]^{\mathrm{T}}$ and $\mathbf{Z}_{\mathbf{X}} = [\mathbf{R}\ \mathbf{P}]^{\mathrm{T}}$. The operator $^{\boxslash}$ removes repeated rows, much like the operator $^{\oslash}$ from Section 5.3 removes repeated columns.

Proof of Theorem 5.1. The inequality states that $\tilde{\mathbf{T}}^{\oslash}$ should not have more columns than rows; hence, $\tilde{\mathbf{T}}^{\oslash}$ is assumed to be a tall matrix. Note that $\mathbf{Q}$ in (5.9) is a unitary matrix. To prove that $\tilde{\mathbf{T}}^{\oslash} = \mathbf{Q}\mathbf{Z}_{\mathbf{X}}^{\boxslash\,\mathrm{T}}$ has full column rank, it is thus sufficient to prove that $\mathbf{Z}_{\mathbf{X}}^{\boxslash\,\mathrm{T}}$ has full column rank or, equivalently, that $\mathbf{Z}_{\mathbf{X}}^{\boxslash}$ has full row rank.

From $\mathbf{X} = \mathbf{M}\mathbf{S}$, it can be seen that $\mathbf{Z}_{\mathbf{X}} = \mathbf{B}\mathbf{Z}_{\mathbf{S}}$ with $\mathbf{B} \in \mathbb{C}^{K^6 \times R^6}$ a block-diagonal matrix containing the blocks $\mathbf{M} \otimes \mathbf{M}^\star \otimes \mathbf{M} \otimes \mathbf{M}^\star$ and $\mathbf{M} \otimes \mathbf{M}^\star$ on the diagonal. Furthermore, it can be seen that $\mathbf{Z}_{\mathbf{X}}^{\boxslash} = \mathbf{B}^{\otimes}\mathbf{Z}_{\mathbf{S}}^{\boxslash}$. $\mathbf{B}^{\otimes}$ is obtained by removing the non-unique rows and columns from $\mathbf{B}$. As we assume that $\mathbf{M}$ has full rank, $\mathbf{B}^{\otimes}$ has full rank as well. Hence, it remains to show that $\mathbf{Z}_{\mathbf{S}}^{\boxslash}$ has full row rank for generic $\mathbf{S}$.

The latter problem is equivalent to showing that $\mathbf{Z}_{\mathbf{S}}^{\boxslash}$ has full row rank for at least one choice of $\mathbf{S}$. This follows from Lemma 5.1, as $\mathbf{S}$ has full row rank if and only if $\mathbf{S}\mathbf{S}^{\mathrm{T}}$ has a non-zero determinant. Hence, the analytic function mentioned in the lemma is $f(\mathbf{A}) = \det(\mathbf{A}\mathbf{A}^{\mathrm{T}})$.

We look for an example in the set of Vandermonde matrices. Let us assume that $\mathbf{S}$ is a transposed Vandermonde matrix:

$$\mathbf{S} = \begin{bmatrix} 1 & d_1 & \cdots & d_1^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & d_R & \cdots & d_R^{N-1} \end{bmatrix}.$$

Then, $\mathbf{S} \odot \mathbf{S}^\star$ is again a transposed Vandermonde matrix [344]:

$$\mathbf{S} \odot \mathbf{S}^\star = \begin{bmatrix} 1 & d_1 d_1^\star & \cdots & d_1^{N-1} {d_1^\star}^{N-1} \\ 1 & d_1 d_2^\star & \cdots & d_1^{N-1} {d_1^\star}^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & d_R d_R^\star & \cdots & d_R^{N-1} {d_R^\star}^{N-1} \end{bmatrix},$$

with R^2 generators $d_i d_j^\star$ for $1 \leq i, j \leq R$. Similarly, $\mathbf{S} \odot \mathbf{S}^\star \odot \mathbf{S} \odot \mathbf{S}^\star$ is a transposed Vandermonde matrix with generators $d_i d_j^\star d_k d_l^\star$ for $1 \leq i, j, k, l \leq R$. Combined, $\mathbf{Z_S}$ and $\mathbf{Z_S^{\boxslash}}$ are Vandermonde matrices with $R^4 + R^2$ and $\frac{1}{4}R^2(R+1)^2 + R^2$ generators, respectively:

$$\begin{cases} d_i d_j^\star d_k d_l^\star, & 1 \leq i \leq k \leq R \text{ and } 1 \leq j \leq l \leq R, \\ d_i d_j^\star, & 1 \leq i, j \leq R. \end{cases}$$

We can now choose different values d_r such that the $\frac{1}{4}R^2(R+1)^2 + R^2$ generators are distinct. Then, $\mathbf{Z_S^{\boxslash}}$ has full row rank. □

Chapter 6

Tensorlab 3.0 — Numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization

Abstract We give an overview of recent developments in numerical optimization-based computation of tensor decompositions that have led to the release of Tensorlab 3.0 in March 2016 (`www.tensorlab.net`). By careful exploitation of tensor product structure in methods such as quasi-Newton and nonlinear least squares, good convergence is combined with fast computation. A modular approach extends the computation to coupled factorizations and structured factors. Given large datasets, different compact representations (polyadic, Tucker, ...) may be obtained by stochastic optimization, randomization, compressed sensing, etc. Exploiting the representation structure allows us to scale the algorithms for constrained/coupled factorizations to large problem sizes.

Reference This chapter is a slightly adapted version of the article [390]. Changes are limited to layout and representation aspects. The candidate authored Sections 6.2 and 6.3 and was a developer of Tensorlab 3.0.

6.1 Introduction

Central to multilinear algebra are tensors, or multiway arrays of numerical values, and their many types of decompositions such as the canonical polyadic decomposition (CPD), the block term decomposition (BTD) or the multilinear singular value decomposition (MLSVD). Similar to their matrix counterparts, these decompositions can be used to analyze data, compress data, make predictions and much more. The multilinear structures allow more complex relations to be modeled, as has been shown in countless applications not only in signal processing [84, 90, 324], but, among others, also in data analytics and machine learning [10, 260, 328].

Tensorlab [392] is a MATLAB toolbox with as main purpose to provide user-friendly access to a variety of state-of-the-art numerical algorithms and tools for tensor computations. In March 2016, the third version of Tensorlab has been released. This chapter gives a birds-eye overview of some new techniques that have been made available. The overview is by no means exhaustive: a full overview can be found at `www.tensorlab.net`. A number of demos illustrating good Tensorlab practice can be accessed at `www.tensorlab.net/demos`.

We continue this section by explaining the history and philosophy of Tensorlab and by fixing the notations. Section 6.2 discusses the SDF framework from Tensorlab, while Section 6.3 explains the concept of tensorization. Section 6.4 introduces a new algorithm for coupled matrix/tensor factorizations in Tensorlab 3.0. Large-scale approaches are discussed in Section 6.5, with a focus on compression, incompleteness, randomizations and efficient representations.

6.1.1 History and philosophy

The first version of Tensorlab provided state-of-the-art algorithms for the computation of CPDs, BTDs or low multilinear rank approximations (LMRLA) as well as a large number of convenience methods involving tensors. These algorithms are based on the complex optimization toolbox (COT) [331, 335], allowing decompositions of both real and complex datasets and/or variables. In optimization problems, real-valued functions with complex arguments are often split into the real part and the imaginary part, and both problems are solved separately. In contrast, the complex Taylor series expansion can be used to generalize standard real-valued optimization algorithms for complex arguments and data, thereby exploiting inherent structure present in derivatives which would otherwise be ignored [12, 335]. COT leverages this structure and provides generalizations of many standard optimization algorithms.

The alternating least squares (ALS) algorithm is undoubtedly the most popular algorithm for tensor decompositions, mainly because of its simplicity. While it effectively exploits multilinear structures and often provides good results quickly, it is numerically not very sophisticated and it has no proven convergence [332, 370]. In Tensorlab, the main focus lies on more advanced optimization algorithms such as nonlinear least squares (NLS) methods, thereby benefiting from the many good results in numerical optimization, including convergence guarantees. The number of iterations needed is often lower because of the quadratic convergence. The asymptotic cost per iteration of NLS can be reduced to the cost of ALS, although with some larger constants [332]. To achieve this low cost, the multilinear structure is exploited and a preconditioned iterative solver is used to determine the step direction. In particular, in NLS algorithms the system

$$\mathbf{G}\mathbf{p} = -\mathbf{g} \tag{6.1}$$

is solved in every iteration, in which $\mathbf{G}$ is the Gramian of the Jacobian and $\mathbf{g}$ is the gradient. As inverting $\mathbf{G}$ is too expensive, the conjugate gradients (CG) method is used. CG requires only the matrix-vector products $\mathbf{G}\mathbf{p}$ to iteratively solve (6.1). In many tensor decomposition algorithms the multilinear structure can be exploited when computing these products. To reduce the number of CG iterations needed, preconditioning is used, i.e., instead of (6.1) the system

$$\mathbf{M}^{-1}\mathbf{G}\mathbf{p} = -\mathbf{M}^{-1}\mathbf{g} \tag{6.2}$$

is solved, in which the preconditioner $\mathbf{M}$ is an easily invertible matrix chosen such that (6.2) is easier to solve. (More technically, the eigenvalues of $\mathbf{M}^{-1}\mathbf{G}$ are more clustered than those of $\mathbf{G}$.) For tensor problems, a block-Jacobi preconditioner, i.e., a block diagonal approximation to $\mathbf{G}$, is often an effective choice [332]. The combination of low per-iteration cost with quadratic convergence of NLS type methods leads to a fast algorithm. In practice, the algorithms also seem more robust for ill-conditioned problems [332].

Since its official launch in February 2013, Tensorlab has seen two more releases. In January 2014, Tensorlab 2.0 was revealed, including the structured data fusion (SDF) framework as its major feature. SDF allows structured and coupled decompositions of multiple full, sparse or incomplete matrices or tensors. This was inspired by the success of specific dedicated algorithms, each exploiting a particular type of constraint on the factor matrices. SDF allows the user to choose different decompositions, constraints and regularizations and combine these to their liking using SDF's own domain specific language [333]. By leveraging the chain rule for derivatives, parametric constraints can be handled easily: over 40 constraints are included, such as nonnegativity, Toeplitz, polynomial, Kronecker, Vandermonde and matrix multiplication. Different types of regularization can be used to model soft constraints as well.

The most recent release from March 2016, Tensorlab 3.0, introduces tensorization and structured tensors, extends and improves the SDF framework while making it more user-friendly, introduces a number of large-scale algorithms and a new algorithmic family, improves coupled matrix/tensor factorizations, and much more. In the following sections, we discuss a number of these new features in more detail.

6.1.2 Notation

An Nth order tensor $\mathcal{T}$ can be factorized in various ways. The (canonical) polyadic decomposition (CPD) writes the tensor as a (minimal) number of rank-1 terms, each of which is the outer product, denoted by $\otimes$, of N non-zero vectors $\mathbf{a}_r^{(n)}$:

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \cdots \otimes \mathbf{a}_r^{(N)} = \left[\!\left[\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)} \right]\!\right],$$

in which the factor matrix $\mathbf{A}^{(n)}$ contains the vectors $\mathbf{a}_r^{(n)}$ as its columns. The higher-order SVD (HOSVD) or multilinear SVD (MLSVD) can be written as the mode-n tensor-matrix product $\cdot_n$ of a core tensor $\mathcal{S}$ and N factor matrices $\mathbf{U}^{(n)}$:

$$\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}^{(1)} \cdot \cdots \cdot_N \mathbf{U}^{(N)}.$$

The block term decomposition (BTD) writes a tensor as a sum of low-multilinear rank terms:

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{S}^{(r)} \cdot_1 \mathbf{U}^{(r,1)} \cdot \cdots \cdot_N \mathbf{U}^{(r,N)}.$$

A special variant of the BTD is the decomposition into a sum of multilinear rank-$(L_r, L_r, 1)$ terms (LL1):

$$\mathcal{T} = \sum_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^{\mathrm{T}}) \otimes \mathbf{c}_r.$$

An overview of these decompositions is given in Fig. 6.1.

The mode-n unfolding of a tensor $\mathcal{T}$ is denoted by $\mathbf{T}_{(n)}$ and concatenates the mode-n vectors as columns in the matrix $\mathbf{T}_{(n)}$. The element-wise product or Hadamard product, the transpose and the Hermitian transpose are denoted by $*$, $\cdot^{\mathrm{T}}$ and $\cdot^{\mathrm{H}}$, respectively. $\langle \cdot, \cdot \rangle$ denotes the inner product.

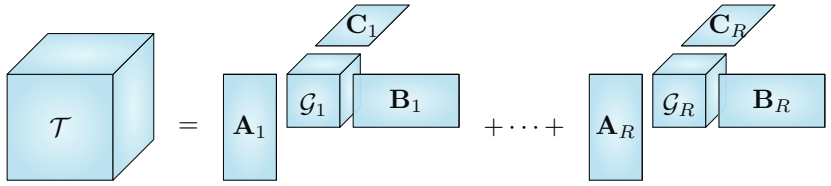


Figure 6.1: Block term decomposition of a tensor $\mathcal{T}$ in terms with multilinear ranks (L_r, M_r, N_r). If $R = 1$, a LMLRA is obtained. If $M_r = L_r$ and $N_r = 1$, thus if the rth core tensor has size $(L_r, L_r, 1)$, a BTD in multilinear rank-$(L_r, L_r, 1)$ terms is obtained. If $L_r = M_r = N_r = 1$, a CPD is obtained.

6.2 Structured data fusion

Structured data fusion (SDF) is a framework for rapid prototyping of analysis and knowledge discovery in one or more multidimensional datasets in the form of tensors. Fig. 6.2 gives a schematic overview. These tensors can be complex, incomplete, sparse and/or structured. Each tensor is decomposed using one of the tensor decompositions that are included in Tensorlab. The factor matrices are possibly shared between the different datasets, meaning that the tensors are coupled. They can also be equal within a tensor decomposition, indicating the presence of symmetry. Furthermore, besides the choice of factorizations, regularization terms can be added as well, based on L_0, L_1 or L_2 norms. Regularization can be used to prevent overfitting but also to implement soft constraints.

In a lot of applications, prior knowledge is available on the factor matrices indicating some kind of structure such as orthogonality or non-negativity. More than 40 structures are readily available in Tensorlab to constrain the factor matrices, cf. Fig. 6.3. Besides the administered structures, a user can design its own constraints as well by providing the mapping and its first-order derivative information. It is worthwhile to note that the constraints are implemented with parametric transformations of underlying optimization variables, rather than with penalty terms. For example, an orthogonal factor matrix of size $I \times R$ requires only $R(I - (R-1)/2)$ variables while a Vandermonde matrix of size $I \times R$ requires only I generating variables. Hence, the solution space is reduced to a restricted search space, and the constraints are imposed exactly rather than only approximately. The chain rule is then internally used to cope with the composition of the tensor decomposition model and the various transformations/constraints, and to solve for the underlying variables.

The type of tensor decomposition, the coupling and the structure imposed on the factors can all be chosen independently of the solver and its options. Two different popular classes of algorithms are available to solve SDF problems in Tensorlab: quasi-Newton (QN) methods and nonlinear least squares (NLS) methods, implemented in `sdf_minf` and `sdf_nls`, respectively. Within the QN methods, both limited memory BFGS (L-BFGS, subdivided in line search and trust region approaches) and nonlinear conjugate gradient (NCG) methods can be selected, while Gauss–Newton (CG-Steihaug and Dogleg trust region approaches) and Levenberg–Marquardt algorithms are implemented within the NLS class.

In Tensorlab 3.0, the SDF framework has been updated in several respects. Two new solvers for symmetric and/or coupled CPDs are introduced (as discussed in Section 6.4), as well as three new factorization types and various updated and new transformations. Besides a focus on content, there has also been a focus on user-friendliness. Using a new language parser (`sdf_check`), it is easier to formulate SDF models and to investigate them. It also helps finding errors in the model. Furthermore, the domain specific language has been made more lenient to allow more flexible model formulation, e.g., by automatically converting arrays to cells and adding braces, wherever necessary.

The handling of incomplete and sparse tensors has also improved from Tensorlab 3.0 on. Note that with the surge of big data applications in mind, the Tensorlab algorithms have a linear time complexity in the number of known/non-zero elements of the data tensor. The SDF features regarding incomplete tensors have shown its value in various applications before, such as in movie recommendation and user participation predictions [333] as well as in the design of alloys and in multidimensional harmonic retrieval [391]. This is further discussed in Section 6.5.2.

6.3 Tensorization

Many powerful tensor tools have been developed throughout the years for analyzing multiway data. When no tensor data is available and only a matrix is given, tensor tools may still be used after first transforming the matrix data to tensor data. This transformation is called tensorization, and many different mappings are possible. The tensorization step is conceptually an important step by itself. Many results concerning tensorization have appeared in the literature in a disparate manner but have not been discussed as such, e.g., [384].

After the tensorization step, one often computes a tensor decomposition. This is especially the case in blind signal separation, where the first tensorization step

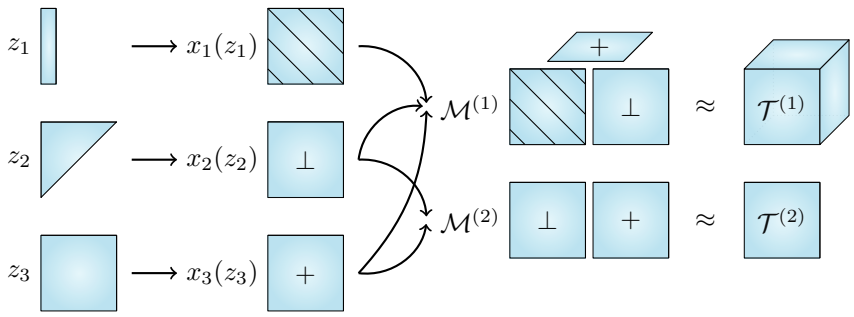


Figure 6.2: A schematic of structured data fusion (adapted from [333]). The vector z_1, upper triangular matrix z_2 (representing a sequence of Householder reflectors) and full matrix z_3 are transformed into a Toeplitz, orthogonal and nonnegative matrix, respectively. The resulting factors are then used to jointly factorize two coupled datasets $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$.

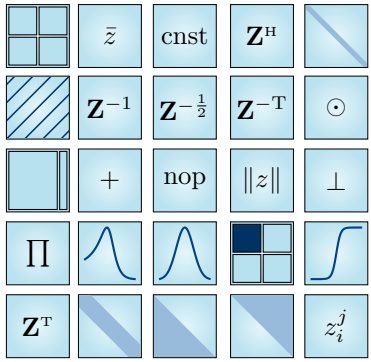


Figure 6.3: More than 40 structures can be imposed on factor matrices in a tensor decomposition using SDF. 25 examples are shown schematically.

implements assumptions on the source signals while the second decomposition step realizes the actual separation of the source signals [118].

Tensorlab 3.0 contains a number of tensorization techniques [118]. Hankelization (Hankel-based mapping) and Löwnerization (Löwner-based mapping) can be used when dealing with approximations by exponentials/sinusoids and rational functions, respectively. Segmentation and decimation are based on folding matrix data, which is, e.g., useful when dealing with large-scale data [51]. Also higher-order and lagged second-order statistics have been included.

Corresponding detensorization techniques have been included where possible. They can be useful, for example, to extract source estimates from the terms in the tensor decomposition. By providing a (noisy) Hankel matrix or tensor for example, the command `dehankelize` returns the averaged anti-diagonals or anti-diagonal slices, respectively.

Tensorization typically involves including redundant information in the higher-order tensor. The number of elements in the obtained tensor can grow quickly, in line with the curse of dimensionality which states that the number of elements in a tensor increases exponentially with the number of dimensions, and so do the computational and memory requirements. To cope with this curse, Tensorlab 3.0 can use efficient representations of the higher-order tensors resulting from the tensorization. The efficiency of these representations can then be exploited in the decomposition algorithms, as discussed in Section 6.5.4.

6.4 Coupled matrix/tensor factorization

Joint decomposition of multiple datasets into rank-1 terms is a common problem in data analysis. Often symmetry constraints are used as well. Both coupling and symmetry, at the level of the data and the factorization, are easy to implement using SDF. In this section, we discuss how the new, specialized coupled and symmetric CPD (CCPD) solver improves convergence and reduces computation time compared to the standard SDF solvers by exploiting both constraints early.

The general SDF solvers `sdf_minf` and `sdf_nls` handle coupling and symmetry by first computing the Gramian of the Jacobian $\mathbf{G}$ and the gradient $\mathbf{g}$ as if no constraints were imposed (see Equation (6.1)). $\mathbf{G}$ and $\mathbf{g}$ are then contracted to $\mathbf{G}_c$ and $\mathbf{g}_c$ which, in this case, boils down to summing the proper blocks, as indicated in Fig. 6.4. The result is a smaller system which is cheaper to solve.

Fig. 6.4 shows that many blocks in $\mathbf{G}$ are repeated because of symmetry. The `ccpd_nls` function takes this into account directly: each unique block is

multiplied by the number of occurrences instead of summing all blocks after computing them. In the case of the gradient, symmetry in the data $\mathcal{T}$ and the decomposition, e.g., $[\![\mathbf{A}, \mathbf{A}, \mathbf{B}]\!]$, has to be considered. In the example, the decomposition is symmetric in the first two modes as the factor matrices are identical. The gradients w.r.t. the first and second mode are only identical if $\mathcal{T}$ is symmetric in the first two modes as well. If this is the case, computing the gradient w.r.t. to second mode is unnecessary. Otherwise, there is no computational gain possible. Detecting symmetry is therefore an important task in the CCPD solvers.

For a regular CPD, a block-Jacobi preconditioner has shown to be effective and efficient to reduce the cost of solving (6.1) because of the Kronecker structure [332]. The `ccpd_nls` algorithm uses a similar preconditioner that exploits symmetry and coupling while keeping the Kronecker structure, in contrast to the non-preconditioned `sdf_nls` algorithm.

To illustrate the performance gain of the new algorithm, consider the following coupled and symmetric problem [127]:

$$\min_{\mathbf{M},\boldsymbol{\kappa}} \left|\left|\mathbf{C}^{(2)} - \mathbf{M}\mathbf{M}^{\mathrm{T}}\right|\right|^2 + \left|\left|\mathcal{C}^{(4)} - [\![\mathbf{M}, \mathbf{M}, \mathbf{M}, \mathbf{M}, \boldsymbol{\kappa}]\!]\right|\right|^2 \tag{6.3}$$

in which $\mathbf{C}^{(2)}$ and $\mathcal{C}^{(4)}$ are constructed using $\mathbf{M} \in \mathbb{R}^{50\times 25}$ and $\boldsymbol{\kappa} \in \mathbb{R}^{1\times 25}$ drawn from a normal distribution. In Table 6.1 the SDF and the NLS algorithms are compared[1]. It is clear that exploiting all symmetry reduces the time per iteration. The block-Jacobi preconditioner used to solve (6.1) improves convergence considerably as can be seen from the reduced number of iterations. The combination of all improvements reduces the total computation time significantly.

6.5 Large-scale tensor decompositions

There exist many strategies for handling large-scale tensors: parallelization of operations, parallel decompositions, incompleteness, compression, exploitation of sparsity and so on. Here, we discuss four techniques readily available in Tensorlab: MLSVD computation using randomized matrix algebra, the use of incomplete tensors and randomized block sampling for polyadic decompositions, and the use of structured tensors.

[1] The timings for both algorithms benefited from a modified version of `mtkrprod` which is not yet released.

Table 6.1: Compared to SDF, CCPD requires less time and fewer iterations to converge when computing (6.3). Increasing the number of CG iterations improves convergence and reduces computation time. All numbers are medians over 50 experiments. Both algorithms use the options `TolX = eps` and `TolFun = eps^2`, with `eps` the machine precision.

	25 CG Iter.		75 CG Iter.	
	SDF	CCPD	SDF	CCPD
Time (s)	70.3	6.9	19.4	6.2
Iterations	170.0	45.5	39.5	29.5
Time/iteration (s)	0.40	0.15	0.47	0.20

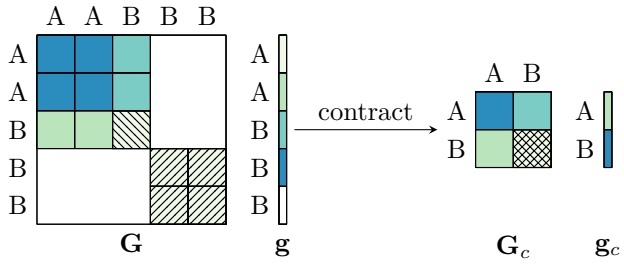


Figure 6.4: Illustration of contraction in `sdf_nls` for a decomposition of $\mathcal{T} = [\![\mathbf{A}, \mathbf{A}, \mathbf{B}]\!]$ and $\mathbf{M} = \mathbf{B}\mathbf{B}^{\mathrm{T}}$. All blocks with the same shading are identical. The CCPD algorithm computes the contracted Gramian and gradient directly.

6.5.1 Randomized compression

Using randomized matrix algebra, we derive a fast yet precise algorithm for computing an approximate multilinear singular value decomposition of a tensor $\mathcal{T}$. The standard way to compute an MLSVD uses the matrix SVD to compute the left singular vectors $\mathbf{U}^{(n)}$, $n = 1, 2, 3$, of the different unfoldings $\mathbf{T}_{(n)}$ of the tensor, and computes the core tensor $\mathcal{S}$ as $\mathcal{T} \cdot_1 \mathbf{U}^{(1)^{\mathrm{T}}} \cdot_2 \mathbf{U}^{(2)^{\mathrm{T}}} \cdot_3 \mathbf{U}^{(3)^{\mathrm{T}}}$ [109]. In very recent literature, one has replaced the SVD by a randomized variant from [180]. Here we present a variant that combines a sequential truncation strategy [377] with randomized SVDs and Q subspace iterations [180]. The full algorithm is described in Algorithm 2.

As example, we create 400 random third-order tensors of size $I_1 \times I_2 \times I_3$ with I_n uniformly distributed in $[100; 400]$ and with multilinear ranks (R_1, R_2, R_3) with R_n distributed uniformly in $[10; 50]$, $n = 1, 2, 3$. The compression size

Algorithm 2: Computation of MLSVD using randomization and subspace iteration. (Implemented as `mlsvd_rsi`.)

Input: Nth order tensor $\mathcal{T}$ of size $I_1 \times \cdots \times I_N$, compression size $R_1 \times \cdots \times R_N$, oversampling parameter P and number of subspace iterations Q.

Output: Factor matrices $\mathbf{U}^{(n)}$, $n = 1, \ldots, N$ and core tensor $\mathcal{S}$ such that $\mathcal{S} \bullet_1 \mathbf{U}^{(1)} \bullet \cdots \bullet_N \mathbf{U}^{(N)} \approx \mathcal{T}$.

Set size $s_n \leftarrow I_n$, $n = 1, \ldots, N$ and $\mathcal{Y} \leftarrow \mathcal{T}$;
for $n = 1 \ldots, N$ **do**
  Let $\mathbf{\Omega}$ be a random matrix of size $\prod_{k \neq n} s_k \times R_n + P$;
  $\mathbf{QR} \xleftarrow{QR} \mathbf{Y}_{(n)}\mathbf{\Omega}$;
  for $q = 1, \ldots, Q$ **do**
    $\mathbf{QR} \xleftarrow{QR} \mathbf{Y}_{(n)}^{\mathrm{T}}\mathbf{Q}$;
    $\mathbf{QR} \xleftarrow{QR} \mathbf{Y}_{(n)}\mathbf{Q}$;
  $\mathbf{USV}^{\mathrm{T}} \xleftarrow{SVD} \mathbf{Q}^{\mathrm{T}}\mathbf{Y}_{(n)}$;
  $\mathbf{U}^{(n)} \leftarrow \mathbf{QU}(:, 1 : J_n)$;
  $s_n \leftarrow R_n + P$;
  $\mathcal{Y} \leftarrow \text{reshape}(\mathbf{SV}^{\mathrm{T}}, s_1, \ldots, s_N)$;
$\mathcal{S} \leftarrow \mathcal{Y}(1 : R_1, \ldots, 1 : R_N)$;

is $(\tilde{R}_1, \tilde{R}_2, \tilde{R}_3)$ with $\tilde{R}_n$ distributed uniformly in $[10; 40]$. The oversampling parameter P is 5 and the number of subspace iterations Q is 2. The relative Frobenius norm error is maximally 4.2% higher in the case of the randomized algorithm `mlsvd_rsi` compared to the standard algorithm `mlsvd`, while the speedup is a factor 3 for small tensors and a factor 25 for larger tensors. If the used compression size is equal to or larger than the multilinear rank of the tensor, the mean relative errors are $1.3 \cdot 10^{-14}$ and $0.5 \cdot 10^{-14}$ for the standard and the randomized algorithm, respectively.

6.5.2 Incomplete tensors

Incomplete tensors occur for two main reasons. First, one can be unable to know some entries, for example, because a sensor breaks down, or because some entries correspond to physically impossible situations, e.g., negative concentrations [391]. In the second case, all elements could be known, but computing or storing all entries is too costly, hence some elements are deliberately omitted. For example, for a rank-R CPD of an Nth order tensor $\mathcal{T}$ of size $I \times \cdots \times I$, the number of entries is I^N, while the number of variables is only NIR. Hence, the number of entries scales exponentially in the order, while the number of

variables scales only linearly. This enables the use of very sparse sampling schemes [391].

Here, we restrict the discussion to the computation of a CPD of an incomplete tensor. Three main techniques can be found in literature [391]. First, unknown elements can be imputed, e.g., by replacing all unknown values with the mean value or with zero. Second, in an expectation-maximization scheme, the unknown values are imputed each iteration with the current best guess from the model. Third, the unknown elements can be ignored altogether. In this last approach, the objective function for a CPD becomes

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \frac{1}{2} ||\mathcal{W} * (\mathcal{T} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!])||_{\mathrm{F}}^2 , \tag{6.4}$$

in which $\mathcal{W}$ is a binary observation tensor. Various optimization schemes have been used to minimize objective (6.4) [9, 333, 359, 388].

Two NLS type algorithms are available in Tensorlab. The first technique scales the Gramian by the fraction of known values, but ignores the structure of the missing data [333]. While this approach is very fast, the result may not be accurate in some cases. If the number of known entries is extremely small, this algorithm may fail [388]. The second technique uses the exact Gramian of the Jacobian, i.e, the structure of the missing data is exploited. Second-order convergence can be achieved, but each iteration is relatively expensive. This is often compensated for, however, as the number of iterations needed for convergence is reduced significantly. As shown in [388], leveraging the exact Gramian can sometimes be crucial in order to find a reasonable solution.

6.5.3 Randomized block sampling

A third technique involves full tensors which may not fit into memory entirely, or for which the computation cost per iteration would be excessive. In [393] a technique called Randomized Block Sampling (RBS) was presented to compute the CPD of large-scale tensors. This method combines block coordinate descent techniques with stochastic optimization as follows. Every iteration, a random subtensor or *block* is sampled from the full tensor. Using this block, one optimization step is performed. Due to the structure of a CPD, only a limited amount of variables are affected in each step. This means multiple steps from multiple blocks can be computed in parallel, as long as the affected variables do not overlap. As only small blocks are used, there is no need to load the full tensor. Blocks can also be generated on-the-fly obfuscating the need to construct a tensor beforehand. Thanks to a simple step restriction schedule, the underlying CP structure can be recovered almost as accurately as if the full tensor were decomposed, even if only a fraction of the data is used.

6.5.4 Efficient representation of structured tensors

Tensors are not always given as a multiway array of numbers or as a list of non-zeros or known entries. The tensor can, for example, be given in the Tucker format as a result of randomized compression, as a Tensor Train approximation [286] to solution of a partial differential equation, or in a Hankel format after tensorization (see Section 6.3). As discussed in [389], the efficient representation of a tensor $\mathcal{T}$ can be exploited by rewriting the objective function

$$\min \|\mathcal{T} - \hat{\mathcal{T}}\|_{\mathrm{F}}^2 \;=\; \min \|\mathcal{T}\|_{\mathrm{F}}^2 - 2\langle \mathcal{T}, \hat{\mathcal{T}} \rangle + \|\hat{\mathcal{T}}\|_{\mathrm{F}}^2, \tag{6.5}$$

in which $\hat{\mathcal{T}}$ can be a CPD, an LMLRA, an LL1 or a BTD. The gradients can be rewritten in a similar way. All norms and inner products at the right-hand side of (6.5), and all matricized tensor times Khatri–Rao or Kronecker products needed for the gradients can be computed efficiently by exploiting the structure of $\mathcal{T}$ and $\hat{\mathcal{T}}$. This can lead to speedups in many algorithms, including ALS, quasi-Newton and NLS algorithms.

Exploiting the structure of tensors by rewriting the objective function as (6.5) does not change the optimization variables. This has as consequence that constraints on factor matrices, symmetry or coupling can be handled trivially. Consider, for example, a nonnegative CPD of a large-scale tensor. Without constraints, the tensor $\mathcal{T}$ is often compressed first to reduce the computational complexity using $\mathcal{S} = \mathcal{T} \cdot_1 \mathbf{U}^{(1)^{\mathrm{T}}} \cdot \cdots \cdot_N \mathbf{U}^{(N)^{\mathrm{T}}}$. The compressed tensor $\mathcal{S}$ is then decomposed as $\left[\!\left[\hat{\mathbf{A}}^{(1)}, \ldots, \hat{\mathbf{A}}^{(N)} \right]\!\right]$. The CPD $[\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$ of $\mathcal{T}$ can be recovered using $\mathbf{A}^{(n)} = \mathbf{U}^{(n)} \hat{\mathbf{A}}^{(n)}$, $n = 1, \ldots, N$. This technique cannot be used to compute the nonnegative CPD, as non-negativity is not preserved by compression, i.e., $\hat{\mathbf{A}}^{(n)} \geq 0$ does not imply $\mathbf{A}^{(n)} \geq 0$ in which $\geq$ holds entry-wise. However, using the structured format, the optimization variables are the full factor matrices $\mathbf{A}^{(n)}$ instead of the compressed ones $\hat{\mathbf{A}}^{(n)}$. Hence standard non-negativity techniques can be used, while still exploiting the structure.

6.6 Conclusion

In this chapter, we have elaborated on a number of features that have been introduced by the third release of Tensorlab in March 2016. First of all, new factorizations and constraints are available in the structured data fusion (SDF) framework. A new tool improves the user-friendliness of SDF by finding model errors early. A number of tensorization and detensorization methods have also been added, allowing the transformation of lower-order to higher-order data, and vice versa. By carefully exploiting the symmetry and coupling structure

in different stages including the preconditioning stage, new solvers for coupled matrix/tensor factorizations have been enabled. Furthermore, a number of new large-scale approaches have been discussed in this chapter, such as randomized block sampling and decompositions of large structured tensors using efficient representations.

Chapter 7

Nonnegative matrix factorization using nonnegative polynomial approximations

Abstract Nonnegative matrix factorization is a key tool in many data analysis applications such as feature extraction, compression and noise filtering. Many existing algorithms impose additional constraints to take into account prior knowledge and to improve the physical interpretation. This chapter proposes a novel algorithm for nonnegative matrix factorization in which the factors are modeled by nonnegative polynomials. Using a parametric representation of finite-interval nonnegative polynomials, we obtain an optimization problem without external non-negativity constraints which can be solved using conventional quasi-Newton or nonlinear least squares methods. The polynomial model guarantees smooth solutions and may realize a noise reduction. A dedicated orthogonal compression enables a significant reduction of the matrix dimensions, without sacrificing accuracy. The overall approach scales well to large matrices. The approach is illustrated with applications in hyperspectral imaging and chemical shift brain imaging.

Reference This chapter is a slightly adapted version of the article [125]. Changes are limited to layout and representation aspects. The candidate performed the research and wrote the article under the guidance of the coauthors.

7.1 Introduction

Matrix factorization techniques compress or analyze a given data matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$ using the bilinear model $\mathbf{X} = \mathbf{W}\mathbf{H}^{\mathrm{T}}$ with $\mathbf{W} \in \mathbb{R}^{I \times R}, \mathbf{H} \in \mathbb{R}^{J \times R}$ and typically $R \ll I, J$. The factor vectors or components in $\mathbf{W}$ and $\mathbf{H}$ can subsequently be used to, e.g., predict missing entries in $\mathbf{X}$ [225] or identify source signals in a blind source separation (BSS) context [85, 89]. Many signals and data are nonnegative by nature such as amplitude spectra, pixel intensities and occurrence counts. Imposing the assumption of non-negativity on $\mathbf{W}$ and $\mathbf{H}$ results in nonnegative matrix factorization (NMF), giving an additive parts-based representation of $\mathbf{X}$ [160]. It is well known that the general problem of NMF does not return unique components [138, 159]. To facilitate the interpretation of the results and to exploit prior knowledge (but not necessarily completely solve the uniqueness problem), many existing NMF techniques impose additional constraints such as sparsity [143, 159, 218], smoothness [77, 137, 408] and orthogonality [81].

Multiple NMF algorithms have been developed using two-block coordinate descent approaches (alternatively solving for one factor matrix while keeping the other matrix fixed, resulting in convex subproblems) such as the multiplicative update (MU) [240], alternating least squares (ALS) [289] and alternating nonnegative least squares methods [217, 239, 372]. Others use a more direct approach by updating both $\mathbf{W}$ and $\mathbf{H}$ in the same step based on projected gradients (PG) [250] or using some parametrization free of non-negativity constraints [82]. It is known that many NMF algorithms are sensitive to initialization and can result in non-smooth solutions [55, 235, 399]. Second, it can be seen that the number of optimization variables $(I + J)R$ can be large for large dimension sizes I, J.

In [82], the parametrizations $\mathbf{W} = \mathbf{E} * \mathbf{E}$ and $\mathbf{H} = \mathbf{F} * \mathbf{F}$ are used, with $\mathbf{E} \in \mathbb{R}^{I \times R}$, $\mathbf{F} \in \mathbb{R}^{J \times R}$ and $*$ the Hadamard product. The technique yields an optimization problem without non-negativity constraints which can be solved with standard quasi-Newton or nonlinear least squares optimization techniques. The approach we propose is also parametrization-based. Rather than parametrizing each variable separately, each entire factor vector (in one or both factor matrices) is represented and approximated by a nonnegative polynomial. This makes sense in various applications, as many (smooth) signals can be well approximated by polynomials with a low degree such as emission or frequency spectra and sinusoidal signals. The approach is related to novel BSS techniques where source and/or mixing vectors are approximated by low-parametric deterministic models such as in [51, 101, 118, 124].

By using a particular parametrization, the complete set of nonnegative

polynomials on a finite interval of interest can be modeled. Smooth results are guaranteed without the use of additional smoothing penalization or regularization terms as in [77, 137], and the number of optimization variables is significantly reduced which is beneficial for large-scale NMF. It is tempting in large-scale NMF to reduce the dimensionality of the problem by a singular value decomposition (SVD). However, the nonnegativity is lost using the orthogonal compression. The nonnegative polynomial approach, on the other hand, does admit a dedicated loss-free orthogonal compression. By carefully exploiting the structure in the resulting problems, efficient algorithms are obtained. These can be used as such or, when polynomials just give a first approximation, to initialize general-purpose NMF algorithms. Note that the technique should not be confused with the fundamentally different polynomial kernel NMF technique from [62].

Notation Parameters, scalars, vectors and matrices are denoted by upper, lower, bold lower and bold upper case characters, e.g., N, a, $\mathbf{a}$ and $\mathbf{A}$, resp. The Hadamard, Kronecker, column-wise Khatri–Rao and row-wise Khatri–Rao products are denoted with $*$, $\otimes$, $\odot$ and $\odot^{\mathrm{T}}$, resp. The columns of $\mathbf{A} \odot \mathbf{B}$ (resp., rows of $\mathbf{A} \odot^{\mathrm{T}} \mathbf{B}$) are the pairwise Kronecker products of the columns (resp., rows) of $\mathbf{A}$ and $\mathbf{B}$. $[\mathbf{A};\ \mathbf{B}]$ denotes the vertical concatenation of matrices $\mathbf{A}$ and $\mathbf{B}$. The transpose, inverse, pseudoinverse, element-wise square and Frobenius norm are denoted by $\cdot^{\mathrm{T}}$, $\cdot^{-1}$, $\cdot^{\dagger}$, $\cdot^{*2}$ and $||\cdot||$, resp.

7.2 Nonnegative polynomial-based NMF

Section 7.2.1 discusses how to model nonnegative polynomials on infinite and finite intervals. These models are then interpreted from a signal processing point of view in Section 7.2.2 and applied to the NMF setting in Section 7.2.3.

7.2.1 Modeling nonnegative polynomials on (in)finite intervals

A degree-D polynomial $f(t)$ is obviously nonnegative for $t \in [-\infty, \infty]$ if it can be written as a squared polynomial, i.e., $f(t) = p(t)^2$. While this representation is not sufficient to model the entire set of nonnegative polynomials on the real axis, the sum-of-squares (SOS) representation is [237]:

$$\forall t \in \mathbb{R} : f(t) \geq 0 \ \Leftrightarrow \ f(t) = \sum\nolimits_{k=1}^{K} p_k(t)^2. \tag{7.1}$$

It is well known that $K = 2$ is sufficient, and each $p_k(t)$ has degree at most $D/2$. Signals of finite rather than infinite length are typically considered in applications, which makes the representation in (7.1) too restrictive for our purposes. This can be understood by considering the class of odd-degree polynomials and more specifically the example $f(t) = t + 2$, for which $f(t) \geq 0$ for $t \in [-1, 1]$ but $f(t) < 0$ for $t < -2$.

A fundamental representation exists in the literature for nonnegative polynomials on a finite interval. We limit the analysis to $t \in [-1, 1]$, which can be generalized to $t \in [a, b]$ using the change in variables $\tilde{f}(t) = f(\frac{2t-(a+b)}{b-a})$. Then [237],[148]:

$$\forall t \in [-1, 1] : f(t) \geq 0 \Leftrightarrow f(t) = f_1(t) + g(t)f_2(t) \tag{7.2}$$

with $g(t) = 1-t^2$ and the degree of $f_1(t)$ and $f_2(t)$ not exceeding $2D$ and $2D-2$, resp. In [237] it is discussed that $f_1(t)$ and $f_2(t)$ should be SOS as in (7.1), but judging from the original work [148, 299, 300] it suffices that $f_1(t)$ and $f_2(t)$ are squared polynomials. Note that the representation is not unique in general, e.g., it is possible that $f_1(t) + g(t)f_2(t) = f_3(t) + g(t)f_4(t)$ for $f_1(t) \neq f_3(t)$ and $f_2(t) \neq f_4(t)$.

7.2.2 Connection with discrete-time signals

Let us first consider the discrete-time signal $\mathbf{p} \in \mathbb{R}^I$ with $p_i = p(t_i)$ for $i = 1, \ldots, I$ and with $p(t)$ a degree-D polynomial. The points t_i do not need to be equidistant. It is clear that one can write $\mathbf{p} = \mathbf{V}\mathbf{c}$ for some evaluated polynomial basis $\mathbf{V} \in \mathbb{R}^{I\times(D+1)}$ and corresponding coefficients $\mathbf{c} \in \mathbb{R}^{D+1}$. For the standard monomial basis, $\mathbf{V}$ has Vandermonde structure. Using a Chebyshev or Legendre basis instead can avoid ill-conditioned situations. An evaluated squared polynomial $\mathbf{f} = \mathbf{p}^{*2}$ then results in $\mathbf{f} = (\mathbf{V}\mathbf{c})^{*2} = (\mathbf{V}\odot^{\mathrm{T}}\mathbf{V})\,(\mathbf{c}\odot\mathbf{c})$. Furthermore, from the representation in (7.2), an evaluated nonnegative polynomial on a finite interval can be represented by

$$\mathbf{f} = (\mathbf{V}\odot^{\mathrm{T}}\mathbf{V})\,(\mathbf{a}\odot\mathbf{a}) + (\mathbf{U}\odot^{\mathrm{T}}\mathbf{U})\,(\mathbf{b}\odot\mathbf{b})\,, \tag{7.3}$$

with matrices $\mathbf{V} \in \mathbb{R}^{I\times(D+1)}, \mathbf{U} \in \mathbb{R}^{I\times D}$, and with coefficient vectors $\mathbf{a} \in \mathbb{R}^{D+1}, \mathbf{b} \in \mathbb{R}^D$. The weighting function $g(t)$ is absorbed in $\mathbf{U}$, i.e., $\forall i, d : u_{id} = \sqrt{1-t_i^2}p_{id}$ with $\mathbf{P} \in \mathbb{R}^{I\times D}$ some evaluated polynomial basis of degree $D-1$.

7.2.3 NMF using nonnegative polynomials

The NMF problem consists of finding the factorization $\mathbf{X} = \mathbf{W}\mathbf{H}^{\mathrm{T}}$ such that $\mathbf{W} \geq 0$ and $\mathbf{H} \geq 0$ with element-wise non-negativity. Assuming that each

factor vector in $\mathbf{W}$ can be well approximated by a finite-interval nonnegative polynomial, the model from (7.3) can be used. With $\mathbf{A} \in \mathbb{R}^{(D+1)\times R}$ and $\mathbf{B} \in \mathbb{R}^{D\times R}$, this results in the following single-sided version:

$$\begin{aligned}\mathbf{X} &= \left[(\mathbf{V}\odot^{\mathrm{T}}\mathbf{V})\,(\mathbf{A}\odot\mathbf{A}) + (\mathbf{U}\odot^{\mathrm{T}}\mathbf{U})\,(\mathbf{B}\odot\mathbf{B})\right]\mathbf{H}^{\mathrm{T}} \\ &= \mathbf{Q}\begin{bmatrix}\mathbf{A}\odot\mathbf{A}\\ \mathbf{B}\odot\mathbf{B}\end{bmatrix}\mathbf{H}^{\mathrm{T}} \quad \text{with } \mathbf{Q} = \begin{bmatrix}\mathbf{V}\odot^{\mathrm{T}}\mathbf{V} & \mathbf{U}\odot^{\mathrm{T}}\mathbf{U}\end{bmatrix}, \end{aligned} \tag{7.4}$$

and $\mathbf{H} \geq 0$. The number of variables in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{H}$ is reduced from $(I+J)R$ to $(2D+1+J)R$. Furthermore, using the model in (7.3) also for the factor vectors in $\mathbf{H}$, one obtains the following double-sided polynomial-based NMF model:

$$\mathbf{X} = \mathbf{Q}^{(w)}\begin{bmatrix}\mathbf{A}\odot\mathbf{A}\\ \mathbf{B}\odot\mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{C}\odot\mathbf{C}\\ \mathbf{D}\odot\mathbf{D}\end{bmatrix}^{\mathrm{T}}\mathbf{Q}^{(h)^{\mathrm{T}}}, \tag{7.5}$$

with $\mathbf{a}_r, \mathbf{b}_r$ and $\mathbf{c}_r, \mathbf{d}_r$ the coefficients related to each factor vector $\mathbf{w}_r$ and $\mathbf{h}_r$, resp., and $\mathbf{Q}^{(w)}, \mathbf{Q}^{(h)}$ similar matrices as in (7.4). The number of variables is reduced to $(2D^{(w)} + 2D^{(h)} + 2)R$, with $D^{(w)}$ (resp., $D^{(h)}$) the degrees of the polynomials used to approximate each factor vector $\mathbf{w}_r$ (resp., $\mathbf{h}_r$).

There exist NMF techniques that make use of a linear model $\mathbf{W} = \mathbf{VA}$ to represent one or both factor matrices, with $\mathbf{V}$ containing some basis functions such as radial basis functions and with $\mathbf{A}$ containing the corresponding coefficients [408]. While the latter methods require explicit non-negativity constraints on $\mathbf{VA}$ and/or $\mathbf{A}$, the constraints are implicitly imposed in (7.5). Second, results of these methods highly depend on the function parameters in $\mathbf{V}$, while the proposed approach only requires predefined degrees $D^{(w)}, D^{(h)}$.

7.3 Algorithms and computational aspects

7.3.1 Quasi-Newton and nonlinear least squares algorithms

Let us consider the single-sided nonnegative polynomial-based NMF (NP-NMF) model in (7.4), with $\mathbf{H} = \mathbf{F}^{*2}$ with $\mathbf{F} \in \mathbb{R}^{J\times R}$ to impose the non-negativity constraint [82]. Given data matrix $\mathbf{X}$, one can find matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{F}$ by minimizing the following least squares objective function:

$$J = \tfrac{1}{2}\left|\left|\mathbf{X} - \mathbf{W}\mathbf{H}^{\mathrm{T}}\right|\right|^2 = \tfrac{1}{2}\left|\left|\mathbf{R}\right|\right|^2, \tag{7.6}$$

with residual matrix $\mathbf{R} = \mathbf{Q}\left[(\mathbf{A}\odot\mathbf{A});(\mathbf{B}\odot\mathbf{B})\right]\left(\mathbf{F}^{*2}\right)^{\mathrm{T}} - \mathbf{X}$.

While NMF methods are typically based on two-block coordinate descent or alternating least squares (ALS, linear convergence) approaches, we consider more direct techniques to minimize (7.6) by updating the estimates in the same iteration. These so-called all-at-once algorithms such as quasi-Newton (QN, superlinear convergence) and nonlinear least squares (NLS, close to quadratic convergence) have been shown to be more robust to overfactoring and to outperform ALS-based algorithms for ill-conditioned cases, although requiring a higher computational load per iteration [9, 335]. For a further discussion, we refer the reader to more detailed literature [278].

QN methods require the expression of the gradient $\mathbf{g}$:

$$\frac{\partial J}{\partial \mathbf{a}_r} = ((\mathbf{I}_{D+1} \otimes \mathbf{a}_r) + (\mathbf{a}_r \otimes \mathbf{I}_{D+1}))^{\mathrm{T}} (\mathbf{V} \odot^{\mathrm{T}} \mathbf{V})^{\mathrm{T}} \mathbf{R} \mathbf{f}_r^{*2} \tag{7.7}$$

$$= 2\mathbf{V}^{\mathrm{T}} \mathrm{diag}(\mathbf{V}\mathbf{a}_r) \mathbf{R} \mathbf{f}_r^{*2}, \qquad r = 1, \ldots, R, \tag{7.8}$$

$$\frac{\partial J}{\partial \mathbf{b}_r} = ((\mathbf{I}_D \otimes \mathbf{b}_r) + (\mathbf{b}_r \otimes \mathbf{I}_D))^{\mathrm{T}} (\mathbf{U} \odot^{\mathrm{T}} \mathbf{U})^{\mathrm{T}} \mathbf{R} \mathbf{f}_r^{*2}$$

$$= 2\mathbf{U}^{\mathrm{T}} \mathrm{diag}(\mathbf{U}\mathbf{b}_r) \mathbf{R} \mathbf{f}_r^{*2}, \qquad r = 1, \ldots, R, \tag{7.9}$$

$$\frac{\partial J}{\partial \mathbf{F}} = 2\mathbf{F} * \left(\mathbf{R}^{\mathrm{T}} \mathbf{Q} \left[(\mathbf{A} \odot \mathbf{A}); (\mathbf{B} \odot \mathbf{B}) \right] \right),$$

in which $\mathrm{diag}(\mathbf{e})$ returns a $I \times I$ diagonal matrix with diagonal $\mathbf{e} \in \mathbb{R}^I$, and in which $\mathbf{I}_I$ is the $I \times I$ identity matrix. To approximate the Hessian by its Gramian $\mathbf{G} = \mathbf{J}^{\mathrm{T}}\mathbf{J}$, NLS methods additionally require the Jacobian matrix $\mathbf{J}$:

$$\partial \,\mathrm{vec}\,(\mathbf{R}) / \partial\, \mathbf{a}_r^{\mathrm{T}} = (\mathbf{f}_r^{*2}) \otimes (\mathrm{diag}(\mathbf{V}\mathbf{a}_r)\mathbf{V}),$$

$$\partial \,\mathrm{vec}\,(\mathbf{R}) / \partial\, \mathbf{b}_r^{\mathrm{T}} = (\mathbf{f}_r^{*2}) \otimes (\mathrm{diag}(\mathbf{W}\mathbf{b}_r)\mathbf{W}),$$

$$\partial \,\mathrm{vec}\,(\mathbf{R}) / \partial\, \mathbf{f}_r^{\mathrm{T}} = \mathbf{I}_N \odot \left(\mathbf{Q} \left[(\mathbf{a}_r \odot \mathbf{a}_r); (\mathbf{b}_r \odot \mathbf{b}_r) \right] \mathbf{f}_r^{\mathrm{T}} \right).$$

These expressions can then be plugged into a numerical optimization solver. Evaluating the objective function and calculating the gradient both cost $\mathcal{O}(I(J + D)R)$ flop, which reduces to $\mathcal{O}(IJR)$ flop as typically $J \gg D$. Rather than constructing and inverting the possibly large Gramian $\mathbf{G}$, fast matrix-vector products $\mathbf{J}^{\mathrm{T}}\mathbf{J}\mathbf{p}$ are used in inexact NLS approaches to iteratively solve the system $\mathbf{G}\mathbf{p} = -\mathbf{g}$. These matrix-vector products and corresponding (block-)diagonal preconditioner can be computed in $\mathcal{O}(IJR)$ flop as well. This order of complexity is equal to that of many standard NMF approaches, e.g., MU or accelerated hierarchical ALS [158, 161, 380]; recall also the convergence

properties mentioned above. In Section 7.3.2, we will further reduce the cost per iteration.

The gradient vector and Jacobian matrix corresponding to the objective function in (7.6) given the double-sided model from (7.5) can be derived in a similar way.

7.3.2 Preprocessing with orthogonal compression

Consider the double-sided NP-NMF model in (7.5). As the typically tall matrices $\mathbf{Q}^{(w)} \in \mathbb{R}^{I\times(2D^{(w)^2}+2D^{(w)}+1)}$ and $\mathbf{Q}^{(h)} \in \mathbb{R}^{J\times(2D^{(h)^2}+2D^{(h)}+1)}$ are known beforehand, a significant compression is possible. Assuming that $I \geq 2D^{(w)}+1$ and $J \geq 2D^{(h)}+1$, it can be shown that $\mathbf{Q}^{(w)}$ and $\mathbf{Q}^{(h)}$ are rank-deficient and have rank $2D^{(w)}+1$ and $2D^{(h)}+1$, resp., independent of the polynomial basis type. Let us compute the corresponding reduced SVDs $\mathbf{Q}^{(w)} = \mathbf{Y}^{(w)}\mathbf{\Sigma}^{(w)}\mathbf{Z}^{(w)^{\mathrm{T}}}$ and $\mathbf{Q}^{(h)} = \mathbf{Y}^{(h)}\mathbf{\Sigma}^{(h)}\mathbf{Z}^{(h)^{\mathrm{T}}}$. Since $\mathbf{Y}^{(w)}$ and $\mathbf{Y}^{(h)}$ are column-wise orthonormal, the double-sided version of (7.6) can without loss of accuracy be expressed as:

$$J = \frac{1}{2}\left\|\tilde{\mathbf{X}} - \left(\mathbf{\Sigma}^{(w)}\mathbf{Z}^{(w)}\right)\begin{bmatrix}\mathbf{A}\odot\mathbf{A}\\ \mathbf{B}\odot\mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{C}\odot\mathbf{C}\\ \mathbf{D}\odot\mathbf{D}\end{bmatrix}^{\mathrm{T}}\left(\mathbf{\Sigma}^{(h)}\mathbf{Z}^{(h)}\right)^{\mathrm{T}}\right\|^2,$$

with $\tilde{\mathbf{X}} = \mathbf{Y}^{(w)^{\mathrm{T}}}\mathbf{X}\mathbf{Y}^{(h)^{\mathrm{T}}} \in \mathbb{R}^{(2D^{(w)}+1)\times(2D^{(h)}+1)}$ the compressed data matrix which needs to be computed only once. While the dimensions are significantly reduced, the Khatri–Rao structure in $\mathbf{Z}^{(w)}, \mathbf{Z}^{(h)}$ is lost. Consequently, we cannot use the simplifications in, e.g., (7.8) and (7.9). Instead, $\partial J/\partial\,\mathbf{a}_r$ is then computed using an expression as in (7.7) but with $\mathbf{V}\odot^{\mathrm{T}}\mathbf{V}$ replaced by the first $(D^{(w)}+1)^2$ columns of $\mathbf{\Sigma}^{(w)}\mathbf{Z}^{(w)}$; like-wise for $\partial J/\partial\,\mathbf{b}_r$, $\partial\,\mathrm{vec}\,(\mathbf{R})/\partial\,\mathbf{a}_r^{\mathrm{T}}$ and $\partial\,\mathrm{vec}\,(\mathbf{R})/\partial\,\mathbf{b}_r^{\mathrm{T}}$. Nevertheless, the complexity is reduced from $\mathcal{O}(IJR)$ to $\mathcal{O}((2D^{(w)}+1)(2D^{(h)}+1)R)$ flop per iteration. This is significant if the number of samples is large or if the components can be well approximated by polynomials of low degree.

7.4 Simulations and applications

In this section, we present some results illustrating the modeling power of nonnegative polynomials for synthetic data and for two real-life applications, together with a timing comparison of the proposed NMF techniques. Unless stated otherwise, the NLS version of NP-NMF with compression step, Chebyshev basis, diagonal preconditioner, random initial coefficients and 50 iterations is

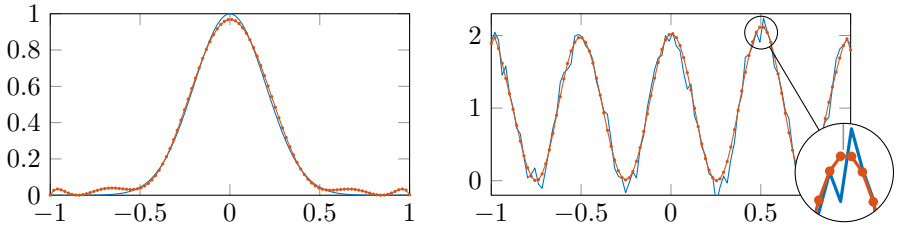


Figure 7.1: A Gaussian function $f(t) = e^{-\frac{1}{2}(5t)^2}$ (left) and a sinusoidal function $f(t) = 1 + \cos(4\pi t)$ with 20 dB SNR (right). Both signals are sampled uniformly 100 times in $[-1, 1]$ (——) and approximated by finite-interval nonnegative polynomials (—•—) of degrees 6 (left) and 10 (right).

used. The polynomial degrees are chosen by visual inspection and with some trial-and-error. Altering the degrees changes the possible number of local extrema and alters the smoothing effect. The complex optimization toolbox [331, 335] is used as numerical optimization solver. Additive Gaussian noise is used. We denote NMF without polynomial structure as 'general NMF'. Unless stated otherwise, it is solved using the accelerated hierarchical ALS (aHALS) technique [86, 161].

A first experiment in Fig. 7.1 considers the approximation of two basic functions. A relative fitting error of 0.047 and 0.092 is obtained, illustrating that nonnegative polynomials can model a variety of shapes and also have a denoising effect.

In a second experiment, we apply NP-NMF to the spectral unmixing problem for non-resolved space object characterization [42, 76, 298]. The spectral signatures (wavelength-dependent absorption features) of space object materials (endmembers) such as aluminum, mylar, paint and silicon from solar cells mix together in the spectral reflectance measurements of the entire object. Given a hyperspectral data matrix $\mathbf{X}$ with I spectral bands and J spectral measurements, the goal is to identify the different endmembers in $\mathbf{W}$ and fractional abundances in $\mathbf{H}$ using the NMF model $\mathbf{X} = \mathbf{W}\mathbf{H}^{\mathrm{T}}$. Let us consider the real-life spectral signatures of $R = 4$ endmembers in the 0.6-1.8 µm range with $I = 120$ spectral bands in Fig. 7.2 [298]. The spectral measurements in $\mathbf{X}$ are constructed by using a matrix $\mathbf{H} \in \mathbb{R}^{4\times 4}$ with entries drawn uniformly from [0,1]. Noise is added to $\mathbf{X}$ in the simulations with a signal-to-noise ratio (SNR) of 40 dB [313]. The median relative fitting error ϵ_f is reported on the noiseless $\mathbf{W}\mathbf{H}^{\mathrm{T}}$ across 10 runs with different initializations. The single-sided NP-NMF method is applied with degrees 6 and 12, each with 30 iterations ($\epsilon_f = 0.0062$ and 0.0049, resp.).

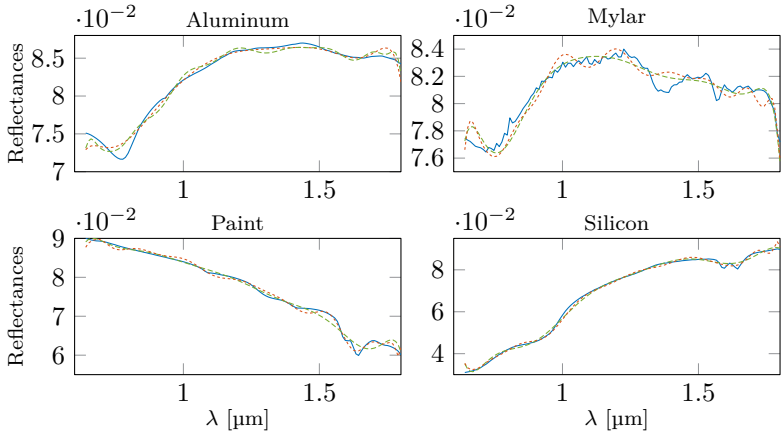


Figure 7.2: The endmembers signatures in the second experiment (——) and aligned NP-NMF estimates using degrees 6 (- - - -) and degrees 12 (·····).

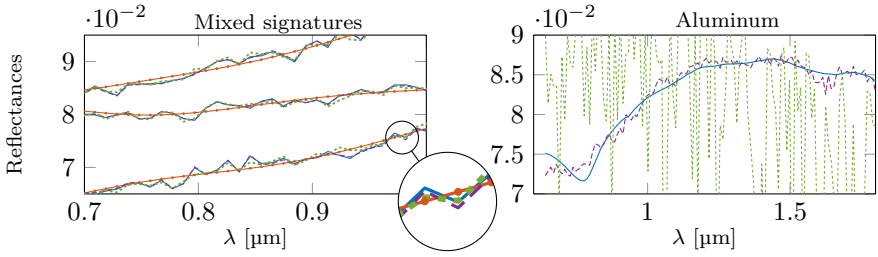


Figure 7.3: (Left) Zoom-in on the mixed signatures in $\mathbf{X}$, with the original signatures (——), fitted NP-NMF signatures with $D^{(w)} = 12$ (—•—) and fitted aHALS signatures. For the latter, two initializations for $\mathbf{W}$ are used: the same initial $\mathbf{W}$ as in the NP-NMF experiment, i.e., a degree-12 nonnegative polynomial with random coefficients (- - - -), and a random nonnegative $\mathbf{W}$ without polynomial structure (·····). (Right) Aluminum signatures estimated by aHALS, to be compared with the NP-NMF estimates in Fig. 7.2 (top, left).

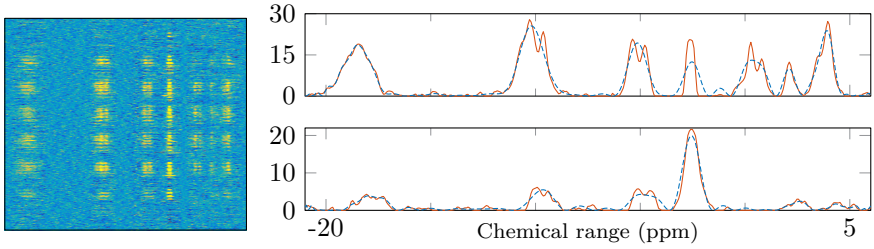


Figure 7.4: (Left) Visualization of $\mathbf{X}^{\mathrm{T}}$ from the third experiment. (Right) Aligned estimated components using NP-NMF (------) and aHALS (——).

General NMF is applied with two types of initializations and 300 iterations ($\epsilon_f = 0.0088$ and 0.0079, resp.). Fig. 7.3 (left) shows that NP-NMF yields smoother estimates than general NMF and is less prone to noise fitting.

Without additional assumptions or constraints, the NMF components cannot be uniquely determined [138, 159]. To be able to compare the estimates, we determine the matrix $\mathbf{Q} \in \mathbb{R}^{R \times R}$ such that $\tilde{\mathbf{W}}\mathbf{Q}$ approximates $\mathbf{W}$ in optimal least-square sense, with $\tilde{\mathbf{W}}$ the estimate of $\mathbf{W}$. The aligned estimates $\tilde{\mathbf{W}}\mathbf{Q}$ obtained from NP-NMF are given in Fig. 7.2; for comparison, the aligned estimated aluminum signature obtained from general NMF is given in Fig. 7.3. The signatures are well approximated by low-degree nonnegative polynomials, i.e., the latter constitute a good class of models. Second, although general NMF yields small fitting errors ϵ_f, the component estimates can be highly non-smooth and difficult to interpret.

A third experiment considers ^{31}P chemical shift imaging data of the human brain [280, 311, 313]. The data set consists of $J = 512$ spectra measured on a $8 \times 8 \times 8$ grid in the brain, each containing $I = 369$ resonance bands between approx. -20 and 5 ppm. Both the single-sided NP-NMF ($D^{(w)} = 34$) and general NMF are applied with $R = 2$ components (brain and muscle). Fig. 7.4 shows similar estimated components, with NP-NMF yielding smooth approximations of the general NMF components. As discussed in the literature, additional constraints (e.g., local sparsity) are needed to incorporate prior knowledge and separate the brain and muscle components.

A fourth experiment compares the timing characteristics of the different methods in the double-sided setting, each with 20 iterations. The columns of $\mathbf{W}$ and $\mathbf{H}$ contain degree-20 nonnegative polynomials. The SNR is 10 dB, $R = 2$ and $D^{(w)}, D^{(h)} = 20$. Fig. 7.5 shows timing results for varying dimensions of $\mathbf{X}$. It can be seen that the compression-based QN and NLS NP-NMF methods

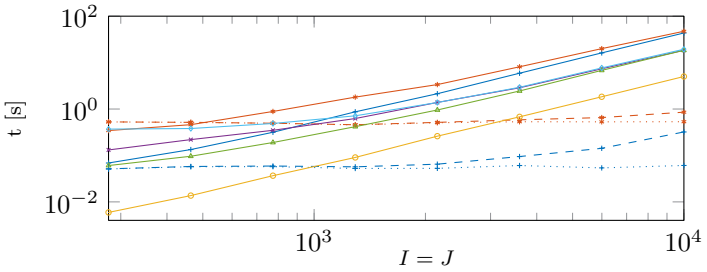


Figure 7.5: Median timing results across 10 simulations for the fourth experiment, for aHALS (—○—), PG [250] (—×—), CG NP-NMF (—+—) and NLS NP-NMF (—•—). The solid and dashed lines for NP-NMF show the uncompressed and compressed versions, respectively. The dotted lines correspond to only the optimization part of the compressed versions, without the compression itself. The element-wise parametrization $\mathbf{W} = \mathbf{E} * \mathbf{E}$ and $\mathbf{H} = \mathbf{F} * \mathbf{F}$ is also considered [82], with CG (—▲—) and NLS (—◆—) versions implemented using the structured data fusion framework from Tensorlab [333, 392].

are faster than aHALS and PG [250] for large dimensions, with the cost of the optimization steps independent of I, J. We note that, in general, NLS needs fewer iterations than CG or ALS to obtain the same accuracy; Fig. 7.5 essentially compares the cost per iteration.

7.5 Conclusion

This chapter proposes novel NMF methods by modeling the components using finite-interval nonnegative polynomials. The parametric representation avoids explicit non-negativity constraints and guarantees smooth solutions. Single- and double-sided variants have been discussed, and quasi-Newton and nonlinear least squares algorithms have been derived. We have indicated that nonnegative polynomials can model a variety of shapes, and the models have successfully been applied on two real-life datasets. A particular orthogonal compression has been proposed that can significantly speed up the NP-NMF methods and can outperform state-of-the-art methods. As future work, the algorithms can be adapted in a spline-like manner to obtain approximating polynomials of lower degrees in different regions of the components.

Chapter 8

Conclusion

8.1 Contributions

In this thesis, we have developed a framework for tensorization (Chapter 2), a Löwner-based tensorization technique suitable for the separation of rational functions (Chapter 3), a segmentation-based tensorization technique suitable for large-scale separation (Chapter 4), an analytical technique for the separation of multi-modulus signals (Chapter 5), structure-exploiting and large-scale algorithms and a new version of Tensorlab (Chapter 6) and a nonnegative polynomial-based method for nonnegative matrix factorization (NMF) which scales well to larger dimensions (Chapter 7). In the following, we give a chapter-by-chapter overview of our contributions. These are categorized according to conceptual contributions (▬), theoretical foundations (▬), algorithms and computational aspects (▬) and proof-of-concepts and real-life applications (▬).

Chapter 2:

An overview of various tensorization techniques. A wide range of tensorization techniques from various domains such as signal processing, graph analysis and machine learning have been discussed and organized in a meaningful framework.

Connections between tensorization techniques. Among other connections, we have established relations between segmentation and time–frequency and time–scale methods, between higher-order statistics and mixed discriminants through polarization, and between the analytical constant modulus algorithm and other algebraic variety-based methods. The connections allowed us to

derive interesting properties, such as low-rank properties for time–frequency & time–scale tensorization methods.

Links between tensors and well-known problems and concepts. Multilinear functions can be bijectively represented by tensors, allowing matrix–matrix multiplication and compound matrices to be connected with tensors. Furthermore, a tensor can also uniquely represent a (multivariate) polynomial. This equivalence between global polynomial optimization on the one hand, and the best symmetric rank-1 tensor approximation on the other hand readily follows. Furthermore, Taylor series, the polynomial Waring problem and polynomial systems can be expressed using tensors.

Application of tensorization in a variety of domains. We have discussed a wide range of applications such as blind signal separation (BSS), graph clustering, data clustering (such as topic modeling and document clustering) and pole estimation.

Chapter 3

A Löwner-based BSS technique. We have proposed a tensorization technique using Löwner matrices called Löwnerization that can be used in BSS. More specifically, Löwnerization allows the separation of rational functions or signals that can be approximated by rational functions. The separation is obtained using a multilinear rank-$(L_r, L_r, 1)$ decomposition of the Löwner tensor.

Connections between rational functions and low-rank Löwner matrices, and factorizations of Löwner matrices. We have shown that rational functions of degree D lead to Löwner matrices of rank D. A Löwner matrix constructed from a rational function also admits a so-called Cauchy decomposition. Each factor matrix is then Cauchy structured. We have investigated rational functions with non-coinciding poles and with coinciding poles, and polynomials.

Uniqueness conditions for Löwner-based separation. We have derived specific uniqueness conditions for Löwner-based BSS based on existing conditions of the multilinear rank-$(L_r, L_r, 1)$ decomposition. The theoretical results show furthermore that polynomials indeed cannot be separated.

Connections between Hankel- and Löwner-based tensorization. A one-to-one relationship exists between the set of Hankel matrices of size $I \times J$ and the set of Löwner matrices of the same size. This allowed us to make a connection between Hankelization and Löwnerization. Although the standard form of Hankelization requires equidistantly sampled points, this is not a requirement for Löwnerization.

Löwner-based separation of maternal and fetal ECG signals. Rational functions can model a variety of shapes, and are well-known for their pole-like behavior. The latter explains why a low-degree rational function can well approximate a heartbeat. This has allowed us to successfully apply Löwner-based BSS succesfully in the context of separating maternal and fetal ECG signals.

Identification of chemical analytes using only a single excitation-emission matrix. Conventional tensor-based methods in spectroscopy stack a collection of excitation–emission matrices corresponding to different mixtures of chemical analytes. The Beer–Lambert law then allows the use of a canonical polyadic decomposition (CPD) on the tensor such that the concentrations of the chemical analytes can be recovered. As spectra can be well approximated by low-degree rational functions, we have been able to show how Löwnerization can be used to separate mixtures based on only a single mixture.

Chapter 4

The contributions in this chapter have followed from a close collaboration with M. Boussé.

A large-scale segmentation-based BSS technique. In large-scale applications, signals and systems often admit a compact representation. This is exploited in the context of (large-scale) BSS. The technique is applied on the source level, the mixing level as well as on both simultaneously. The latter gives rise to twofold segmentation. Furthermore, also higher-order segmentation is applied, in which a vector is reshaped into a higher-order tensor. This enables an even more compact representation.

Connections between segmentation and Hankel-based tensorization and further low-rank properties. The columns of a matrix obtained from segmentation are a subset of the columns of a matrix obtained from Hankelization. Hence, segmentation can be seen as a compact version of Hankelization. Furhermore,

signals that yield low-rank Hankel matrices also yield low-rank matrices after segmentation. However, the reverse is not true. Periodic signals, for example, can lead to low-rank matrices after segmentation. It is discussed that the rank of a matrix after segmentation of a signal has rank R if and only if the signal can be written as a sum of R (and not less than R) Kronecker product vectors.

A decomposition in (rank-L_r $\otimes$ vector) terms and a butterfly decomposition. Higher-order segmentation and twofold segmentation both lead to two novel decompositions. Algorithms and uniqueness conditions have been provided, partly also in the follow-up paper of Chapter 4 [53].

Segmentation-based separation of maternal and fetal ECG signals. Various signals admit a compact representation such as ECG signals. This allows the separation of maternal and fetal ECG signals, much like Löwnerization has allowed in Chapter 3.

Segmentation-based direction of arrival (DOA) estimation. Estimating the DOA of signals is an important goal in array processing. Given uniform linear arrays (ULA), segmentation can be used to extract the direction of arrival of simultaneously impinging signals. Because of the compact representation, the technique scales well to large-scale ULAs. Besides the far-field setting, the technique lends itself well to near-field and multipath cases also. The proposed technique is compared with other DOA methods such as ESPRIT and MUSIC.

Chapter 5

An analytical blind separation technique of multi-modulus signals. The analytical constant modulus algorithm (ACMA) allows the separation of constant modulus signals such as 4-QAM signals [384]. We have developed a more general method that also allows the separation of multi-modulus signals.

Uniqueness conditions for analytical multi-modulus separation. We have derived working assumptions for the separation of multi-modulus signals. More specifically, the technique applies a (coupled) CPD after segmentation of the null space of a constructed matrix which we briefly denote as $\mathbf{T}$. We have shown that for a set of generic source signals, a tall matrix $\mathbf{T}$ has full column rank.

A generalized rank-1 detection procedure. The requirement of a tall matrix $\mathbf{T}$ can cause some limitations, as the number of rows and columns of $\mathbf{T}$ depend strictly on the number of signals and source signals, respectively. We have generalized a previously developed rank-1 detection procedure. Given a space spanned by a number of Kronecker-structured vectors, the original formulation of the procedure allowed the recovery of these vectors. While it does not recover Kronecker-structured vectors from a space spanned by such vectors and other arbitrary vectors, the generalization of the procedure does, provided some algebraic conditions apply.

Alternatively formulated, the generalization allows the algebraic recovery of the factor vectors $\mathbf{a}_r$ and $\mathbf{b}_r$ from a given tensor $\mathcal{T}$ which admits the following expression:

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r + \sum_{q=1}^{Q} \mathbf{D}_q \otimes \mathbf{e}_q.$$

This can be seen as a CPD model in the presence of interfering terms. Note that there exists an upper bound on Q in function of R and in function of the dimensions of $\mathcal{T}$, and that additional algebraic conditions apply. This new procedure has allowed us to conjecture a new relaxed lower bound on the number of samples required for multi-modulus separation.

A generalization to blind deconvolution of multi-modulus signals. Using two different approaches, we have extended the method from instantaneous separation to convolutive separation in a BSI context.

Separation of CPSK and 16-QAM signals. The techniques have been showcased in a context of CPSK and 16-QAM separation, as well as compared with other state-of-the-art techniques in function of both signal-to-noise ratio and number of samples available. Unlike the other algorithms, the proposed method does provide a perfect recovery in the noiseless case and does work well when only a limited amount of samples are available.

Chapter 6

The contributions in this chapter have followed from a close collaboration with N. Vervliet.

Tensorization techniques. We have developed a framework of tensorization techniques within Tensorlab containing novel tensorization techniques such

as Hankelization, Löwnerization, segmentation and decimation as well as already available techniques such as fourth-order cumulants and stacked lagged covariance matrices. Furthermore, a detensorization technique has been provided for each deterministic tensorization technique. The techniques are flexible such that vectors and matrices can be tensorized to tensors of various orders, dimensions, etc.

Large-scale and efficient representations. We have presented a number of recently developed approaches for large-scale tensor decompositions. Furthermore, related to tensorization, we have allowed the decomposition of tensors based on efficient tensor representations. In this way, one avoids the construction of a tensor at all, bypassing the curse of dimensionality.

Extended and improved version of structured data fusion (SDF). The new version of SDF is more efficient by the addition of a number of new solvers, and is also more user-friendly with the addition of a language parser and the inclusion of a more lenient domain specific language. This is mainly contributed by N. Vervliet.

Coupled matrix/tensor factorization algorithms. The joint decomposition of multiple datasets is a common problem in data analysis. The new solvers with updated preconditioners have improved convergence and have reduced computation time, exploiting as much symmetry and coupling as possible. This is mainly contributed by N. Vervliet.

A new version of Tensorlab. These new developments and algorithms have been made available in Tensorlab 3.0 [392]. Together with the release, the web page has been renewed, and an online manual and a set of demonstrations have been added. These demonstrations illustrate the use of Tensorlab in various applications such as DOA estimation, independent component analysis (ICA), independent vector analysis (IVA) and user-involvement prediction, among others [126].

Chapter 7

A nonnegative matrix factorization (NMF) technique based on nonnegative polynomials. Polynomials can be used well to approximate various types of signals. By parametrizing a nonnegative polynomial in a finite interval using a closed-form expression, we have shown how these nonnegative polynomials

can be used in a context of NMF to reduce the number of variables and to provide a denoising effect.

Large-scale NMF techniques and algorithms. We have successfully shown how the polynomial-based NMF method allows orthogonal compression such that a significant dimension reduction is obtained. Together with the previous nonnegative-based parametrization, this has enabled the method to scale well to larger problem dimensions. Unlike the proposed technique, it needs to be pointed out that conventional orthogonal compression based on, e.g., SVD or QR, is not suitable in an NMF context as it destroys the nonnegativity.

Chemical shift brain imaging and the separation of hyperspectral endmember signatures. The method has been showcased in a context of chemical shift brain imaging and in a setting of hyperspectral imaging. In the latter application, we have seen that the proposed model readily provides much smoother solutions than state-of-the-art general NMF methods without the need of smoothing constraints.

8.2 Prospective work

The remaining part of this thesis provides a number of pointers regarding prospective work. Besides the permanent objective of uncovering additional tensorization techniques and related tensor properties, we would suggest the following major directions:

- *Focus on other tensor decompositions.* The bulk of the tensorization techniques in this thesis have been presented in combination with mainly the CPD and the block term decomposition (BTD). Some have also been discussed in the context of tensor trains (TT), such as segmentation or quantization. However, it is evident that many more tensorization techniques can be developed that are useful in combination with the TT, hierarchical Tucker and more general tensor networks, among others.
- *Neural networks and other machine learning tools.* Neural networks with a single hidden layer have recently been connected with tensorization and tensor decompositions [204, 205]. Interesting results can be expected as well for a larger number of layers and for deep networks. Second, further results concerning data clustering and graph clustering, e.g., in the direction of spectral clustering, might be obtained. Other tools from machine learning such as dimensionality reduction techniques, supervised learning methods and prediction algorithms might strongly benefit from

tensorization as well. It does not come as a surprise that Google has recently released a part of its machine learning software library as open source under the name of TensorFlow [1].

- *Spline-like tensorization decomposition.* A number of signal models have been discussed, such as exponential polynomials, rational functions, nonnegative polynomials and compact representations using segmentation. Each of these models can be used to approximate specific types of signals. The approximation of signals using different models in different regions could be a further extension. This brings the excitingly new field of coupled tensor decompositions to the forefront [333]. In the case of coupled CPDs, relaxed uniqueness conditions and algebraic algorithms have already been developed [345, 346].

We would also like to propose a number of minor suggestions:

- *Time–frequency & time–scale transformations.* These transformations have been applied a number of times, especially in the domains of audio processing and biomedical signal processing. However, they have appeared rather disparately, without significant theoretical results on the validity of their use. Although Section 2.6.5 contained a discussion on these methods with a number of new results, an overview indicating the use of specific transformations in specific situations could provide a welcome theoretical foundation. For example, the influence of the type of window function or wavelet should be investigated in more detail.

- *Algebraic variety-based methods.* Section 2.7.2 has discussed tensorization techniques which stack a number of constructed matrices. While the obtained tensor does not readily admit a CPD, a manipulation can be performed to obtain one or more coupled CPDs. This technique has appeared in the context of deterministic BSS (of which the separation of multi-modulus signals from Chapter 5 is an example) and in combination with algebraic CPDs where none of the factor matrices have full column rank. It is expected that this group of tensorization techniques has many more capabilities.

- *Polynomial-based nonnegative tensor factorization (NTF).* We have discussed the use of nonnegative polynomials for NMF, but these nonnegative polynomials could also be applied in the context of NTF [85]. By applying the provided orthogonal compression, a significant dimension reduction can be obtained for large-scale tensors.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*. 2015. URL: `http://tensorflow.org/` (p. 206).

[2] K. Abed-Meraim, W. Qiu, and Y. Hua. "Blind system identification". In: *Proceedings of the IEEE* 85.8 (1997), pp. 1310–1322 (pp. 4, 146).

[3] S. Abrar. "A family of reduced-constellation algorithms for blind equalization of square-QAM signals". In: *Proceedings of the IEEE International Conference on Microelectronics (ICM)*. 2005, pp. 296–300 (p. 146).

[4] S. Abrar, A. Roy, and J. Axford. "Sliced multi-modulus blind equalization algorithm". In: *ETRI Journal* 27.3 (2005), pp. 257–266 (p. 147).

[5] F. Abrard and Y. Deville. "A time–frequency blind signal separation method applicable to underdetermined mixtures of dependent sources". In: *Signal Processing* 85.7 (2005), pp. 1389–1403 (p. 61).

[6] E. Acar, R. Bro, and A. Smilde. "Data fusion in metabolomics using coupled matrix and tensor factorizations". In: *Proceedings of the IEEE* 103.9 (Sept. 2015), pp. 1602–1620 (pp. 28, 148, 155).

[7] E. Acar, S. Çamtepe, M. Krishnamoorthy, and B. Yener. "Modeling and multiway analysis of chatroom tensors". In: *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*. Vol. 3495. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, pp. 256–268 (pp. 29, 30).

[8] E. Acar, D. Dunlavy, and T. Kolda. "A scalable optimization approach for fitting canonical tensor decompositions". In: *Journal of Chemometrics* 25.2 (2011), pp. 67–86 (p. 29).

[9] E. Acar, D. Dunlavy, T. Kolda, and M. Mørup. "Scalable tensor factorizations for incomplete data". In: *Chemometrics and Intelligent Laboratory Systems* 106.1 (2011), pp. 41–56 (pp. 184, 192).

[10] E. Acar and B. Yener. "Unsupervised multiway data analysis: A literature survey". In: *IEEE Transactions on Knowledge and Data Engineering* 21.1 (2009), pp. 6–20 (p. 174).

[11] E. Acar, C. Aykut–Bingol, H. Bingol, R. Bro, and B. Yener. “Multiway analysis of epilepsy tensors”. In: *Bioinformatics* 23.13 (2007), pp. i10–i18 (pp. 61, 65).

[12] T. Adalı, P. J. Schreier, and L. L. Scharf. “Complex-valued signal processing: The proper way to deal with impropriety”. In: *IEEE Transactions on Signal Processing* 59.11 (2011), pp. 5101–5125 (p. 174).

[13] M. Alier, M. Felipe, I. Hernández, and R. Tauler. “Trilinearity and component interaction constraints in the multivariate curve resolution investigation of NO and O_3 pollution in Barcelona”. In: *Analytical and Bioanalytical Chemistry* 399.6 (2011), pp. 2015–2029 (p. 51).

[14] M. Alier, M. Felipe-Sotelo, I. Hernández, and R. Tauler. “Variation patterns of nitric oxide in Catalonia during the period from 2001 to 2006 using multivariate data analysis methods”. In: *Analytica Chimica Acta* 642.1 (2009), pp. 77–88 (p. 51).

[15] J. B. Allen and L. R. Rabiner. “A unified approach to short-time Fourier analysis and synthesis”. In: *Proceedings of the IEEE* 65.11 (1977), pp. 1558–1564 (p. 61).

[16] A. L. F. de Almeida, G. Favier, and L. R. Ximenes. “Space–time–frequency (STF) MIMO communication systems with blind receiver based on a generalized PARATUCK2 model”. In: *IEEE Transactions on Signal Processing* 61.8 (2013), pp. 1895–1909 (p. 62).

[17] A. Anandkumar, R. Ge, D. Hsu, and S. Kakade. “A tensor spectral approach to learning mixed membership community models”. In: *Proceedings of the Conference on Learning Theory*. 2013, pp. 867–881 (pp. 86, 90).

[18] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. “Tensor decompositions for learning latent variable models.” In: *Journal of Machine Learning Research* 15.1 (2014), pp. 2773–2832 (pp. 20, 70, 82).

[19] C. M. Andersen and R. Bro. “Practical aspects of PARAFAC modeling of fluorescence excitation–emission data”. In: *Journal of Chemometrics* 17.4 (2003), pp. 200–215 (pp. 30, 113).

[20] C. Andersson and R. Bro. “The *N*-way Toolbox for MATLAB”. In: *Chemometrics and Intelligent Laboratory Systems* 52.1 (2000), pp. 1–4 (p. 29).

[21] A. C. Antoulas and B. D. O. Anderson. “On the scalar rational interpolation problem”. In: *IMA Journal of Mathematical Control and Information* 3.2-3 (1986), pp. 61–88 (pp. 53, 95, 99).

[22] A. C. Antoulas, A. C. Ionita, and S. Lefteriu. “On two-variable rational interpolation”. In: *Linear Algebra and its Applications* 436.8 (2012), pp. 2889–2915 (p. 52).

[23] A. C. Antoulas and D. C. Sorensen. “Approximation of large-scale dynamical systems: An overview”. In: *International Journal of Applied Mathematics and Computer Science* 11 (2001), pp. 1093–1121 (pp. 52, 95).

[24] T. Antoulas. “A tutorial introduction to the Loewner framework for model reduction”. 9th Elgersburg Workshop Mathematische Systemtheorie. 2014 (p. 95).

[25] L. Atzori, A. Iera, and G. Morabito. “The Internet of Things: A survey”. In: *Computer networks* 54.15 (2010), pp. 2787–2805 (p. 1).

[26] F. Ayres, L. G. Díez, and Á. G. Vázquez. *Matrices*. McGraw–Hill New York, 1962 (p. 40).

[27] R. Badeau and R. Boyer. “Fast multilinear singular value decomposition for structured tensors”. In: *SIAM Journal on Matrix Analysis and Applications* 30.3 (2008), pp. 1008–1021 (p. 49).

[28] B. W. Bader, T. G. Kolda, et al. MATLAB *Tensor Toolbox version 2.5*. Jan. 2012. URL: `http://www.sandia.gov/~tgkolda/TensorToolbox/` (p. 29).

[29] K. Batselier, P. Dreesen, and B. D. Moor. “On the null spaces of the Macaulay matrix”. In: *Linear Algebra and its Applications* 460 (2014), pp. 259–289 (p. 43).

[30] H. Becker, L. Albera, P. Comon, R. Gribonval, F. Wendling, and I. Merlet. “Brain-source imaging: From sparse to tensor models”. In: *IEEE Signal Processing Magazine* 32.6 (Nov. 2015), pp. 100–112 (p. 61).

[31] H. Becker, P. Comon, L. Albera, M. Haardt, and I. Merlet. “Multi-way space–time–wave-vector analysis for EEG source separation”. In: *Signal Processing* 92.4 (2012), pp. 1021–1031 (p. 61).

[32] V. Belevitch. “Interpolation matrices”. In: *Philips Research Reports* 25 (1970), pp. 337–369 (pp. 53, 99).

[33] A. Belouchrani, K. Abed–Meraim, J. Cardoso, and E. Moulines. “A blind source separation technique using second-order statistics”. In: *IEEE Transactions on Signal Processing* 45.2 (1997), pp. 434–444 (pp. 7, 20, 27, 72).

[34] A. Belouchrani, K. Abed-Meraim, J. Cardoso, and E. Moulines. “Second-order blind separation of temporally correlated sources”. In: *Proceedings of the International Conference on Digital Signal Processing*. Citeseer. 1993, pp. 346–351 (p. 72).

[35] A. Belouchrani and M. G. Amin. “Blind source separation based on time–frequency signal representations”. In: *IEEE Transactions on Signal Processing* 46.11 (1998), pp. 2888–2897 (p. 72).

[36] A. Belouchrani, M. G. Amin, N. Thirion–Moreau, and Y. D. Zhang. “Source separation and localization using time–frequency distributions: An overview”. In: *IEEE Signal Processing Magazine* 30.6 (2013), pp. 97–107 (p. 71).

[37] J. Benesty and P. Duhamel. “Fast constant modulus adaptive algorithm”. In: *IEE Proceedings F (Radar and Signal Processing)* 138.4 (1991), pp. 379–387 (p. 146).

[38] J. Bennett, S. Lanning, et al. “The Netflix prize”. In: *Proceedings of KDD cup and workshop*. New York, NY, USA. 2007, pp. 3–6 (pp. 3, 7, 30).

[39] J. Bennett. *Collected Papers of R.A. Fisher*. Vol. 2. University of Adelaide Press, 1972, pp. 351–354 (p. 82).

[40] A. R. Benson, D. F. Gleich, and J. Leskovec. “Higher-order organization of complex networks”. In: *Science* 353.6295 (2016), pp. 163–166 (p. 20).

[41] A. R. Benson, D. F. Gleich, and J. Leskovec. “Tensor spectral clustering for partitioning higher-order network structures”. In: *Proceedings of the SIAM International Conference on Data Mining*. 2015, pp. 118–126 (pp. 85, 86).

[42] M. Berry, M. Browne, A. Langville, V. Pauca, and R. Plemmons. “Algorithms and applications for approximate nonnegative matrix factorization”. In: *Computational Statistics and Data Analysis* 52.1 (2007), pp. 155–173 (p. 194).

[43] A. Bertrand. “Distributed signal processing for wireless EEG sensor networks”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 23.6 (Dec. 2015), pp. 923–935 (pp. 119, 130).

[44] G. Beylkin and M. Mohlenkamp. “Algorithms for numerical analysis in high dimensions”. In: *SIAM Journal on Scientific Computing* 26.6 (2005), pp. 2133–2159 (p. 32).

[45] G. Beylkin, J. Garcke, and M. J. Mohlenkamp. “Multivariate regression and machine learning with sums of separable functions”. In: *SIAM Journal on Scientific Computing* 31.3 (2009), pp. 1840–1857 (p. 32).

[46] H. N. Bharath, D. M. Sima, N. Sauwen, U. Himmelreich, L. De Lathauwer, and S. Van Huffel. “Non-negative canonical polyadic decomposition for tissue type differentiation in gliomas”. In: *IEEE Journal of Biomedical and Health Informatics* (2016) (p. 70).

[47] D. M. Blei, A. Y. Ng, and M. I. Jordan. “Latent Dirichlet allocation”. In: *Journal of Machine Learning Research* 3 (Jan. 2003), pp. 993–1022 (p. 70).

[48] D. L. Boley, F. T. Luk, and D. Vandevoorde. “A fast method to diagonalize a Hankel matrix”. In: *Linear Algebra and its Applications* 284.1 (1998), pp. 41–52 (p. 101).

[49] C. Bourin and P. Bondon. “Efficiency of high-order moment estimates”. In: *IEEE Transactions on Signal Processing* 46.1 (1998), pp. 255–258 (p. 84).

[50] M. Boussé, O. Debals, and L. De Lathauwer. “A novel deterministic method for large-scale blind source separation”. In: *Proceedings of the 23rd IEEE European Signal Processing Conference (EUSIPCO)*. Aug. 2015, pp. 1890–1894 (pp. 119, 130).

[51] M. Boussé, O. Debals, and L. De Lathauwer. “A tensor-based method for large-scale blind source separation using segmentation”. In: *IEEE Transactions on Signal Processing* 65.2 (Jan. 2017), pp. 346–358 (pp. 21, 28, 47, 51, 55, 56, 89, 117, 180, 188).

[52] M. Boussé, O. Debals, and L. De Lathauwer. “A tensor-based method for large-scale blind system identification using segmentation”. In: *Proceedings of the 24th IEEE European Signal Processing Conference (EUSIPCO)*. Aug. 2016, pp. 2015–2019 (p. 56).

[53] M. Boussé, O. Debals, and L. De Lathauwer. “Tensor-based large-scale blind system identification using segmentation”. Technical Report 16–109, ESAT–STADIUS, KU Leuven, Belgium. Sept. 2016 (p. 202).

[54] M. Boussé, G. Goovaerts, N. Vervliet, O. Debals, S. Van Huffel, and L. De Lathauwer. “Irregular heartbeat classification using Kronecker product equations”. In: *Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. To appear. July 2017 (p. 52).

[55] C. Boutsidis and E. Gallopoulos. “SVD-based initialization: A head start for nonnegative matrix factorization”. In: *Pattern Recognition* 41.4 (2008), pp. 1350–1362 (p. 188).

[56] R. Bro. “Multi-way analysis in the food industry. Models, algorithms, and applications.” PhD thesis. University of Amsterdam (NL) & Royal Veterinary and Agricultural University (DK), 1998 (pp. 7, 8, 113).

[57] R. Bro. “PARAFAC. Tutorial and applications”. In: *Chemometrics and intelligent laboratory systems* 38.2 (1997), pp. 149–171 (pp. 30, 57, 113).

[58] R. Bro and H. Kiers. “A new efficient method for determining the number of components in PARAFAC models”. In: *Journal of Chemometrics* 17.5 (2003), pp. 274–286 (p. 30).

[59] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. "Class-based n-gram models of natural language". In: *Computational Linguistics* 18.4 (1992), pp. 467–479 (pp. 78, 79).

[60] A. M. Bruckstein, D. L. Donoho, and M. Elad. "From sparse solutions of systems of equations to sparse modeling of signals and images". In: *SIAM Review* 51.1 (2009), pp. 34–81 (pp. 6, 61).

[61] B. Buchberger. "Gröbner bases and applications". In: *London Mathematical Society Lecture Note Series*. Ed. by B. Buchberger and F. Winkler. Vol. 251. Cambridge, 1998 (p. 43).

[62] I. Buciu, N. Nikolaidis, and I. Pitas. "Nonnegative matrix factorization in polynomial feature space". In: *IEEE Transactions on Neural Networks* 19.6 (June 2008), pp. 1090–1100 (p. 189).

[63] C. F. Caiafa and A. Cichocki. "Multidimensional compressed sensing and their applications". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3.6 (Nov. 2013), pp. 355–380 (p. 118).

[64] V. D. Calhoun, T. Adalı, G. D. Pearlson, and K. A. Kiehl. "Neuronal chronometry of target detection: Fusion of hemodynamic and event-related potential data". In: *Neuroimage* 30.2 (2006), pp. 544–553 (p. 94).

[65] D. Callaerts. "Signal separation methods based on singular value decomposition and their application to the real-time extraction of the fetal electrocardiogram from cutaneous recordings". PhD thesis. KU Leuven, Dec. 1989 (pp. 114, 140).

[66] E. Candès and Y. Plan. "Matrix completion with noise". In: *Proceedings of the IEEE* 98.6 (2010), pp. 925–936 (pp. 3, 24).

[67] E. Candès and M. Wakin. "An introduction to compressive sampling". In: *IEEE Signal Processing Magazine* 25.2 (2008), pp. 21–30 (p. 118).

[68] J.-F. Cardoso, S. Bose, and B. Friedlander. "Output cumulant matching for source separation". In: *IEEE Signal Processing / ATHOS Workshop on Higher-Order Statistics*. 1995, pp. 44–50 (p. 84).

[69] J.-F. Cardoso. "Source separation using higher order moments". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1989, pp. 2109–2112 (p. 83).

[70] J.-F. Cardoso. "Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1991, pp. 3109–3112 (pp. 16, 77, 84).

[71] J.-F. Cardoso and A. Souloumiac. "Blind beamforming for non-Gaussian signals". In: *IEE Proceedings F — Radar and Signal Processing* 140.6 (1993), pp. 362–370 (pp. 12, 20, 21, 27, 83).

[72] J. D. Carroll and J.-J. Chang. “Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart–Young decomposition”. In: *Psychometrika* 35.3 (1970), pp. 283–319 (pp. 8, 20, 29).

[73] R. B. Cattell. “Parallel proportional profiles and other principles for determining the choice of factors by rotation”. In: *Psychometrika* 9.4 (1944), pp. 267–283 (p. 20).

[74] R. B. Cattell. “The three basic factor-analytic research designs — Their interrelations and derivatives.” In: *Psychological bulletin* 49.5 (1952), p. 499 (p. 20).

[75] A. Cayley. “XXVIII — On the theory of the analytical forms called trees”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 13.85 (1857), pp. 172–176 (p. 39).

[76] C.-I. Chang. “An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis”. In: *IEEE Transactions on Information Theory* 46.5 (Aug. 2000), pp. 1927–1932 (pp. 30, 194).

[77] Z. Chen, A. Cichocki, and T. M. Rutkowski. “Constrained non-negative matrix factorization method for EEG analysis in early detection of Alzheimer disease”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 5. 2006, pp. 896–896 (pp. 188, 189).

[78] L. Chiantini and G. Ottaviani. “On generic identifiability of 3-tensors of small rank”. In: *SIAM Journal on Matrix Analysis and Applications* 33.3 (2012), pp. 1018–1037 (pp. 26, 121).

[79] L. Chiantini, G. Ottaviani, and N. Vannieuwenhoven. “An algorithm for generic and low-rank specific identifiability of complex tensors”. In: *SIAM Journal on Matrix Analysis and Applications* 35.4 (2014), pp. 1265–1287 (pp. 26, 121).

[80] H.-I. Choi and W. J. Williams. “Improved time–frequency representation of multicomponent signals using exponential kernels”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.6 (1989), pp. 862–871 (p. 63).

[81] S. Choi. “Algorithms for orthogonal nonnegative matrix factorization”. In: *IEEE International Joint Conference on Neural Networks*. 2008, pp. 1828–1832 (p. 188).

[82] M. Chu, F. Diele, R. Plemmons, and S. Ragni. “Optimality, computation and interpretation of nonnegative matrix factorizations”. In: *SIAM Journal on Matrix Analysis* (2004), pp. 4–8030 (pp. 188, 191, 197).

[83] A. Cichocki and S. Amari. *Adaptive blind signal and image processing: learning algorithms and applications*. Vol. 1. John Wiley, 2002 (pp. 94, 118).

[84] A. Cichocki, D. Mandic, A. H. Phan, C. Caiafa, G. Zhou, Q. Zhao, and L. De Lathauwer. "Tensor decompositions for signal processing applications: From two-way to multiway component analysis". In: *IEEE Signal Processing Magazine* 32.2 (2015), pp. 145–163 (pp. 8, 20, 25, 95, 120, 133, 149, 174).

[85] A. Cichocki, R. Zdunek, A. Phan, and S. Amari. *Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation*. Wiley, 2009 (pp. 94, 95, 188, 206).

[86] A. Cichocki, R. Zdunek, and S.-I. Amari. "Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization". In: *Proceedings of the International Conference on International Conference on Independent Component Analysis and Signal Separation (ICA)*. Vol. 4666. Lecture Notes in Computer Science. Springer. 2007, pp. 169–176 (p. 194).

[87] L. Cohen. "Time–frequency distributions — A review". In: *Proceedings of the IEEE* 77.7 (1989), pp. 941–981 (pp. 14, 21, 60, 63).

[88] N. Cohen, O. Sharir, and A. Shashua. "On the expressive power of deep learning: A tensor analysis". In: *Proceedings of the 29th Annual Conference on Learning Theory*. Ed. by V. Feldman, A. Rakhlin, and O. Shamir. Vol. 49. Proceedings of Machine Learning Research. June 2016, pp. 698–728 (pp. 20, 90).

[89] P. Comon and C. Jutten. *Handbook of blind source separation: Independent component analysis and applications*. Academic Press, 2010 (pp. 4, 6, 14, 71, 79, 83, 94, 118, 123, 130, 146, 188).

[90] P. Comon. "Tensors: A brief introduction". In: *IEEE Signal Processing Magazine* 31.3 (2014), pp. 44–53 (p. 174).

[91] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain. "Symmetric tensors and symmetric tensor rank". In: *SIAM Journal on Matrix Analysis and Applications* 30.3 (2008), pp. 1254–1279 (p. 40).

[92] P. Comon and B. Mourrain. "Decomposition of quantics in sums of powers of linear forms". In: *Signal Processing* 53.2 (1996), pp. 93–107 (p. 40).

[93] P. Comon and M. Rajih. "Blind identification of under-determined mixtures based on the characteristic function". In: *Signal Processing* 86.9 (2006), pp. 2271–2281 (p. 73).

[94] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi. "Tensor decomposition of EEG signals: A brief review". In: *Journal of Neuroscience Methods* 248 (2015), pp. 59–69 (p. 30).

[95] N. Courtois, A. Klimov, J. Patarin, and A. Shamir. “Efficient algorithms for solving overdefined systems of multivariate polynomial equations”. In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2000, pp. 392–407 (p. 43).

[96] I. Daubechies et al. *Ten lectures on wavelets*. Vol. 61. SIAM, 1992 (pp. 63, 65).

[97] I. Daubechies. “The wavelet transform, time–frequency localization and signal analysis”. In: *IEEE Transactions on Information Theory* 36.5 (1990), pp. 961–1005 (pp. 21, 63).

[98] S. Daumont and D. Le Guennec. “An analytical multimodulus algorithm for blind demodulation in a time-varying MIMO channel context”. In: *International Journal Of Digital Multimedia Broadcasting* 2010, 307927 (2010), pp. 1–11 (pp. 147, 170).

[99] L. De Lathauwer. “A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization”. In: *SIAM Journal on Matrix Analysis and Applications* 28.3 (2006), pp. 642–666 (pp. 16, 29, 77, 148, 157, 158, 160).

[100] L. De Lathauwer. “Algebraic techniques for the blind deconvolution of constant modulus signals”. In: *Proceedings of the 12th IEEE European Signal Processing Conference (EUSIPCO)*. 2004, pp. 225–228 (pp. 146, 162).

[101] L. De Lathauwer. “Blind separation of exponential polynomials and the decomposition of a tensor in rank-$(L_r, L_r, 1)$ terms”. In: *SIAM Journal on Matrix Analysis and Applications* 32.4 (2011), pp. 1451–1474 (pp. 9, 14, 28, 47, 48, 55, 56, 94, 95, 101, 106, 107, 112, 118, 122, 124, 126, 137, 188).

[102] L. De Lathauwer. “Block component analysis, a new concept for blind source separation”. In: *Latent Variable Analysis and Signal Separation*. Vol. 7191. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2012, pp. 1–8 (pp. 56, 95, 112, 118, 122).

[103] L. De Lathauwer. “Decompositions of a higher-order tensor in block terms — Part I: Lemmas for partitioned matrices”. In: *SIAM Journal on Matrix Analysis and Applications* 30.3 (2008), pp. 1022–1032 (pp. 9, 27, 118, 122).

[104] L. De Lathauwer. “Decompositions of a higher-order tensor in block terms — Part II: Definitions and uniqueness”. In: *SIAM Journal on Matrix Analysis and Applications* 30.3 (2008), pp. 1033–1066 (pp. 9, 27, 95, 97, 106, 109, 118, 122, 133).

[105] L. De Lathauwer. “Signal processing based on multilinear algebra”. PhD thesis. Belgium: Katholieke Universiteit Leuven, 1997 (p. 97).

[106] L. De Lathauwer and A. de Baynast. “Blind deconvolution of DS-CDMA signals by means of decomposition in rank-$(1, L, L)$ terms”. In: *IEEE Transactions on Signal Processing* 56.4 (2008), pp. 1562–1571 (p. 122).

[107] L. De Lathauwer, J. Castaing, and J.-F. Cardoso. “Fourth-order cumulant-based blind identification of underdetermined mixtures”. In: *IEEE Transactions on Signal Processing* 55.6 (2007), pp. 2965–2973 (pp. 95, 101).

[108] L. De Lathauwer and B. De Moor. “On the blind separation of non-circular sources”. In: *Proceedings of the 11th IEEE European Signal Processing Conference (EUSIPCO)*. 2002, pp. 1–4 (p. 72).

[109] L. De Lathauwer, B. De Moor, and J. Vandewalle. “A multilinear singular value decomposition”. In: *SIAM Journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1253–1278 (pp. 9, 97, 182).

[110] L. De Lathauwer, B. De Moor, and J. Vandewalle. “A prewhitening-induced bound on the identification error in independent component analysis”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 52.3 (2005), pp. 546–554 (p. 84).

[111] L. De Lathauwer, B. De Moor, and J. Vandewalle. “Fetal electrocardiogram extraction by blind source subspace separation”. In: *IEEE Transactions on Biomedical Engineering* 47.5 (2000), pp. 567–572 (pp. 4, 114, 140).

[112] L. De Lathauwer, B. De Moor, and J. Vandewalle. “On the best rank-1 and rank-$(R_1, R_2, \ldots, R_n)$ approximation of higher-order tensors”. In: *SIAM Journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1324–1342 (pp. 9, 27, 40, 97).

[113] L. De Lathauwer and D. Nion. “Decompositions of a higher-order tensor in block terms — Part III: Alternating least squares algorithms”. In: *SIAM Journal on Matrix Analysis and Applications* 30.3 (2008), pp. 1067–1083 (pp. 9, 27, 95, 97, 109).

[114] L. De Lathauwer and J. Vandewalle. “Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, \ldots, R_N)$ reduction in multilinear algebra”. In: *Linear Algebra and its Applications* 391 (2004), pp. 31–55 (pp. 8, 27).

[115] B. De Moor. “The singular value decompositions and long and short spaces of noisy matrices”. In: *IEEE Transactions on Signal Processing* 41.9 (Sept. 1993), pp. 2826–2838 (p. 137).

[116] M. De Vos, D. Nion, S. Van Huffel, and L. De Lathauwer. “A combination of parallel factor and independent component analysis”. In: *Signal Processing* 92.12 (2012), pp. 2990–2999 (pp. 14, 89).

[117] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmini, and W. Van Paesschen. "Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone". In: *NeuroImage* 37.3 (2007), pp. 844–854 (pp. 30, 65).

[118] O. Debals and L. De Lathauwer. "Stochastic and deterministic tensorization for blind signal separation". In: *Latent Variable Analysis and Signal Separation*. Vol. 9237. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, Aug. 2015, pp. 3–13 (pp. 45, 118, 180, 188).

[119] O. Debals and L. De Lathauwer. "Stochastic and deterministic tensorization for blind signal separation". In: *Latent Variable Analysis and Signal Separation*. Vol. 9237. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2015, pp. 3–13 (p. 146).

[120] O. Debals and L. De Lathauwer. "The concept of tensorization". Technical Report 17–99, ESAT–STADIUS, KU Leuven, Belgium. Aug. 2017 (p. 19).

[121] O. Debals, L. De Lathauwer, and M. Van Barel. "About higher-order Löwner tensors". Technical Report 17–98, ESAT–STADIUS, KU Leuven, Belgium. Apr. 2017 (p. 54).

[122] O. Debals, M. Sohail, and L. De Lathauwer. "Analytical multi-modulus algorithms based on coupled canonical polyadic decompositions". Technical Report 16–150, ESAT–STADIUS, KU Leuven, Belgium. Sept. 2016 (pp. 77, 145).

[123] O. Debals, M. Van Barel, and L. De Lathauwer. "Blind signal separation of rational functions using Löwner-based tensorization". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2015, pp. 4145–4149 (p. 95).

[124] O. Debals, M. Van Barel, and L. De Lathauwer. "Löwner-based blind signal separation of rational functions with applications". In: *IEEE Transactions on Signal Processing* 64.8 (Apr. 2016), pp. 1909–1918 (pp. 14, 53–57, 93, 118, 137, 188).

[125] O. Debals, M. Van Barel, and L. De Lathauwer. "Nonnegative matrix factorization using nonnegative polynomial approximations". In: *IEEE Signal Processing Letters* 24.7 (July 2017), pp. 948–952 (p. 187).

[126] O. Debals, F. Van Eeghem, N. Vervliet, and L. De Lathauwer. "Tensorlab Demos - Release 3.0". Technical Report 16–68, ESAT–STADIUS, KU Leuven, Belgium. Mar. 2016 (p. 204).

[127] O. Debals, N. Vervliet, F. Van Eeghem, and L. De Lathauwer. "Using coupled tensor decompositions and incomplete tensors for independent component analysis". Technical Report 16–171, ESAT–STADIUS, KU Leuven, Belgium. 2016 (p. 181).

[128] W. Deburchgraeve, P. Cherian, M. De Vos, R. Swarte, J. Blok, G. H. Visser, P. Govaert, and S. Van Huffel. "Neonatal seizure localization using PARAFAC decomposition". In: *Clinical Neurophysiology* 120.10 (2009), pp. 1787–1796 (p. 65).

[129] W. Ding, L. Qi, and Y. Wei. "Fast Hankel tensor–vector product and its application to exponential data fitting". In: *Numerical Linear Algebra with Applications* 22.5 (2015), pp. 814–832 (pp. 48, 49).

[130] I. Domanov and L. De Lathauwer. "Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition". In: *SIAM Journal on Matrix Analysis and Applications* 35.2 (2014), pp. 636–660 (pp. 29, 77, 88, 121).

[131] I. Domanov and L. De Lathauwer. "Generic uniqueness conditions for the canonical polyadic decomposition and INDSCAL". In: *SIAM Journal on Matrix Analysis and Applications* 36.4 (2015), pp. 1567–1589 (pp. 26, 121).

[132] I. Domanov and L. De Lathauwer. *Generic uniqueness of a structured matrix factorization and applications in blind source separation.* Tech. rep. 14-153. ESAT–STADIUS, KU Leuven (Leuven, Belgium), 2014 (p. 107).

[133] I. Domanov and L. De Lathauwer. "Generic uniqueness of a structured matrix factorization and applications in blind source separation". In: *IEEE Journal of Selected Topics in Signal Processing* 10.4 (June 2016), pp. 701–711 (p. 121).

[134] I. Domanov and L. De Lathauwer. "On the uniqueness of the canonical polyadic decomposition of third-order tensors — Part I: Basic results and uniqueness of one factor matrix". In: *SIAM Journal on Matrix Analysis and Applications* 34.3 (2013), pp. 855–875 (pp. 8, 26, 121, 155).

[135] I. Domanov and L. De Lathauwer. "On the uniqueness of the canonical polyadic decomposition of third-order tensors — Part II: Uniqueness of the overall decomposition". In: *SIAM Journal on Matrix Analysis and Applications* 34.3 (2013), pp. 876–903 (pp. 8, 26, 121, 149, 155, 170).

[136] I. Domanov and L. De Lathauwer. "Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm". In: *Linear Algebra and its Applications* 513 (2017), pp. 342–375 (pp. 8, 26, 29, 121).

[137] Q. Dong and L. Li. "Smooth incomplete matrix factorization and its applications in image/video denoising". In: *Neurocomputing* 122 (2013), pp. 458–469 (pp. 188, 189).

[138] D. Donoho and V. Stodden. "When does non-negative matrix factorization give a correct decomposition into parts?" In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 17. MIT Press, 2004, pp. 1141–1148 (pp. 188, 196).

[139] P. Dreesen, M. Ishteva, and J. Schoukens. "Decoupling multivariate polynomials using first-order information and tensor decompositions". In: *SIAM Journal on Matrix Analysis and Applications* 36.2 (2015), pp. 864–879 (pp. 14, 36).

[140] P. Dreesen, M. Schoukens, K. Tiels, and J. Schoukens. "Decoupling static nonlinearities in a parallel Wiener–Hammerstein system: A first-order approach". In: *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. 2015, pp. 987–992 (p. 36).

[141] G. Easley, D. Labate, and W.-Q. Lim. "Sparse directional image representations using the discrete shearlet transform". In: *Applied and Computational Harmonic Analysis* 25.1 (2008), pp. 25–46 (p. 65).

[142] C. Eckart and G. Young. "The approximation of one matrix by another of lower rank". In: *Psychometrika* 1.3 (1936), pp. 211–218 (pp. 5, 126, 137).

[143] J. Eggert and E. Korner. "Sparse coding and NMF". In: *IEEE Proceedings of International Joint Conference on Neural Networks* 4 (2004), pp. 2529–2533 (p. 188).

[144] E. Eidinger and A. Yeredor. "Blind source separation via the second characteristic function with asymptotically optimal weighting". In: *Proceedings of the 23rd IEEE Convention of Electrical and Electronics Engineers*. 2004, pp. 404–407 (p. 84).

[145] M. Elad, P. Milanfar, and G. H. Golub. "Shape from moments — An estimation theory perspective". In: *IEEE Transactions on Signal Processing* 52.7 (2004), pp. 1814–1829 (p. 49).

[146] Y. C. Eldar and G. Kutyniok. *Compressed sensing: Theory and applications*. Cambridge University Press, 2012 (p. 123).

[147] European Commission. *Mergers: Commission fines Facebook €110 million for providing misleading information about WhatsApp takeover*. Press Release. May 18, 2017. URL: `http://europa.eu/rapid/press-release_IP-17-1369_en.htm` (p. 2).

[148] M. Fekete. "Proof of three propositions of Paley". In: *Bulletin of the American Mathematical Society* 41.2 (1935), pp. 138–144 (p. 190).

[149] S. Feldmann and G. Heinig. "Vandermonde factorization and canonical representations of block Hankel matrices". In: *Linear Algebra and its Applications* 241 (1996), pp. 247–278 (p. 48).

[150] C. Fernandes, G. Favier, J. Mota, et al. “PARAFAC-based blind identification of convolutive MIMO linear systems”. In: *System Identification*. Vol. 15. 1. 2009, pp. 1704–1709 (p. 89).

[151] A. Ferréol, L. Albera, and P. Chevalier. “Fourth-order blind identification of underdetermined mixtures of sources (FOBIUM)”. In: *IEEE Transactions on Signal Processing* 53.5 (2005), pp. 1640–1653 (pp. 95, 101).

[152] M. Fiedler. “Hankel and Löwner matrices”. In: *Linear Algebra and its Applications* 58 (1984), pp. 75–95 (pp. 53, 54, 99, 102, 108).

[153] J. A. R. Fonollosa. “Sample cumulants of stationary processes: Asymptotic results”. In: *IEEE Transactions on Signal Processing* 43.4 (1995), pp. 967–977 (pp. 14, 84).

[154] S. Fridli, L. Lócsi, and F. Schipp. “Rational function systems in ECG processing”. In: *Computer Aided Systems Theory–EUROCAST 2011*. Springer, 2012, pp. 88–95 (p. 114).

[155] O. Friman, J. Cedefamn, P. Lundberg, M. Borga, and H. Knutsson. “Detection of neural activity in functional MRI using canonical correlation analysis”. In: *Magnetic Resonance in Medicine* 45.2 (2001), pp. 323–330 (p. 72).

[156] X. Gao, O. Edfors, F. Rusek, and F. Tufvesson. “Massive MIMO performance evaluation based on measured propagation data”. In: *IEEE Transactions on Wireless Communications* 14.7 (Mar. 2015), pp. 3899–3911 (p. 130).

[157] J.-L. Gauvain and C.-H. Lee. “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains”. In: *IEEE Transactions on Speech and Audio Processing* 2.2 (1994), pp. 291–298 (p. 70).

[158] N. Gillis. “Nonnegative matrix factorization: Complexity, algorithms and applications”. PhD thesis. UCL, 2011 (p. 192).

[159] N. Gillis. “Sparse and unique nonnegative matrix factorization through data preprocessing”. In: *Journal of Machine Learning Research* 13 (2012), pp. 3349–3386 (pp. 188, 196).

[160] N. Gillis. “The why and how of nonnegative matrix factorization”. In: *Regularization, Optimization, Kernels, and Support Vector Machines*. Machine Learning and Pattern Recognition. Chapman & Hall / CRC, 2014. Chap. 12, pp. 257–291 (p. 188).

[161] N. Gillis and F. Glineur. “Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization”. In: *Neural Computation* 24.4 (2012), pp. 1085–1105 (pp. 192, 194).

[162] F. Giri and E.-W. Bai. *Block-oriented nonlinear system identification*. Vol. 1. Springer, 2010 (p. 36).

[163] D. F. Gleich, L.-H. Lim, and Y. Yu. "Multilinear pagerank". In: *SIAM Journal on Matrix Analysis and Applications* 36.4 (2015), pp. 1507–1541 (pp. 8, 85).

[164] D. N. Godard. "Self-recovering equalization and carrier tracking in two-dimensional data communication systems". In: *IEEE Transactions on Communications* 28.11 (1980), pp. 1867–1875 (p. 146).

[165] G. H. Golub and C. F. Van Loan. *Matrix computations*. 4th ed. Johns Hopkins University Press, 2013 (pp. 24, 152).

[166] R. Gooch and J. Lundell. "The CM array: An adaptive beamformer for constant modulus signals". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 11. 1986, pp. 2523–2526 (p. 146).

[167] G. Goovaerts, B. Vandenberk, R. Willems, and S. V. Huffel. "Tensor-based detection of T Wave Alternans using ECG". In: *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Aug. 2015, pp. 6991–6994 (p. 52).

[168] G. Goovaerts, O. De Wel, B. Vandenberk, R. Willems, and S. Van Huffel. "Detection of irregular heartbeats using tensors". In: *Proceedings of the Computing in Cardiology Conference (CinC)*. IEEE. 2015, pp. 573–576 (p. 52).

[169] A. Goupil and J. Palicot. "New algorithms for blind equalization: The constant norm algorithm family". In: *IEEE Transactions on Signal Processing* 55.4 (2007), pp. 1436–1444 (p. 146).

[170] L. Grasedyck. *Polynomial approximation in hierarchical Tucker format by vector tensorization*. Apr. 2010 (pp. 9, 28, 51, 52).

[171] L. Grasedyck, D. Kressner, and C. Tobler. "A literature survey of low-rank tensor approximation techniques". In: *GAMM-Mitteilungen* 36.1 (2013), pp. 53–78 (pp. 29, 118, 123, 127, 144).

[172] O. Grellier and P. Comon. "Analytical blind discrete source separation". In: *Proceedings of the 10th IEEE European Signal Processing Conference (EUSIPCO)*. Sept. 2000, pp. 1–4 (p. 77).

[173] C. Guerrero, A. M. Trigueros, and J. I. Franco. "Time–frequency EEG analysis in epilepsy: What is more suitable?" In: *Proceedings of the 5th IEEE International Symposium on Signal Processing and Information Technology*. 2005, pp. 202–207 (p. 65).

[174] R. C. Gunning and H. Rossi. *Analytic functions of several complex variables*. Vol. 368. American Mathematical Society, 2009 (p. 170).

[175] M. Haardt and J. A. Nossek. "Simultaneous Schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multidimensional harmonic retrieval problems". In: *IEEE Transactions on Signal Processing* 46.1 (1998), pp. 161–169 (p. 27).

[176] M. Haardt, F. Roemer, and G. Del Galdo. "Higher-order svd-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems". In: *IEEE Transactions on Signal Processing* 56.7 (2008), pp. 3198–3213 (p. 49).

[177] W. Hackbusch. *Tensor spaces and numerical tensor calculus*. Vol. 42. Springer Series in Computational Mathematics. Springer Science & Business Media, 2012 (pp. 8, 28, 29, 32).

[178] W. Hackbusch and S. Kühn. "A new scheme for the tensor representation". In: *Journal of Fourier Analysis and Applications* 15.5 (2009), pp. 706–722 (p. 9).

[179] K. Hägglöf, P. O. Lindberg, and L. Svensson. "Computing global minima to polynomial optimization problems using Gröbner bases". In: *Journal of Global Optimization* 7.2 (1995), pp. 115–125 (p. 43).

[180] N. Halko, P. G. Martinsson, and J. A. Tropp. "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions". In: *SIAM Review* 53.2 (2011), pp. 217–288. URL: `http://dx.doi.org/10.1137/090771806` (p. 182).

[181] J. D. Hamilton. *Time series analysis*. Vol. 2. Princeton University Press, 1994 (p. 94).

[182] H. H. Harman. *Modern factor analysis*. third ed. Univ. of Chicago Press, 1976 (p. 6).

[183] R. Harshman. "Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis". In: *UCLA Working Papers in Phonetics* 16 (1970), pp. 1–84 (pp. 8, 20, 29, 30, 154).

[184] R. A. Harshman. "Determination and proof of minimum uniqueness conditions for PARAFAC1". In: *UCLA Working Papers in Phonetics* 22.111-117 (1972), p. 3 (pp. 20, 154).

[185] J. A. Hartigan and M. A. Wong. "Algorithm AS 136: A k-means clustering algorithm". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1 (1979), pp. 100–108 (pp. 70, 86).

[186] S. Haykin. *Neural networks: A comprehensive foundation*. 2nd ed. Prentice Hall, 2004 (p. 87).

[187] L. He, M. Amin, C. Reed Jr, and R. Malkemes. “A hybrid adaptive blind equalization algorithm for QAM signals in wireless communications”. In: *IEEE Transactions on Signal Processing* 52.7 (2004), pp. 2058–2069 (pp. 147, 170).

[188] D. Hilbert and B. Sturmfels. *Theory of algebraic invariants*. Cambridge University Press, 1993 (p. 39).

[189] S. Hill. *From J-Phone to Lumia 1020: A complete history of the camera phone*. Aug. 12, 2013. URL: `https://www.digitaltrends.com/mobile/camera-phone-history/` (p. 2).

[190] F. L. Hitchcock. “Multiple invariants and generalized rank of a p-way matrix or tensor”. In: *Studies in Applied Mathematics* 7.1-4 (1928), pp. 39–79 (p. 20).

[191] F. L. Hitchcock. “The expression of a tensor or a polyadic as a sum of products”. In: *Studies in Applied Mathematics* 6.1-4 (1927), pp. 164–189 (p. 20).

[192] H. Hotelling. “Relations between two sets of variates”. In: *Biometrika* 28.3/4 (1936), pp. 321–377 (p. 72).

[193] D. Hsu and S. M. Kakade. “Learning mixtures of spherical Gaussians: Moment methods and spectral decompositions”. In: *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*. ACM. 2013, pp. 11–20 (pp. 20, 70, 82).

[194] Y. Hua and T. K. Sarkar. “Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38.5 (1990), pp. 814–824 (p. 49).

[195] N. E. Huang and S. S. Shen. *Hilbert–Huang transform and its applications*. Vol. 16. World Scientific, 2014 (p. 66).

[196] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu. “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis”. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 454. 1971. 1998, pp. 903–995 (p. 66).

[197] Y.-D. Huang and M. Barkat. “Near-field multiple source localization by passive sensor array”. In: *IEEE Transactions on Antennas and Propagation* 39.7 (July 1991), pp. 968–975 (p. 142).

[198] K.-C. Hung, D. Lin, and C.-N. Ke. "Variable-step-size multimodulus blind decision-feedback equalization for high-order QAM based on boundary MSE estimation". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 4. 2004, pp. 881–884 (pp. 147, 170).

[199] B. Hunyadi, D. Camps, L. Sorber, W. Van Paesschen, M. De Vos, S. Van Huffel, and L. De Lathauwer. "Block term decomposition for modelling epileptic seizures". In: *EURASIP Journal on Advances in Signal Processing* 2014.1 (2014), pp. 1–19 (pp. 56, 61, 65).

[200] A. Hyvärinen and E. Oja. "Independent component analysis: Algorithms and applications". In: *Neural networks* 13.4 (2000), pp. 411–430 (pp. 94, 118).

[201] A. K. Jain, M. N. Murty, and P. J. Flynn. "Data clustering: A review". In: *ACM Computing Surveys (CSUR)* 31.3 (1999), pp. 264–323 (p. 25).

[202] W. C. Jakes. *Microwave mobile communications*. New York: Wiley, 1974 (p. 130).

[203] C. J. James and C. W. Hesse. "Independent component analysis for biomedical signals". In: *Physiological Measurement* 26.1 (2005), R15–R39 (p. 118).

[204] M. Janzamin, H. Sedghi, and A. Anandkumar. "Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods". In: *arXiv preprint arXiv:1506.08473* (2016) (pp. 20, 87, 205).

[205] M. Janzamin, H. Sedghi, and A. Anandkumar. "Score function features for discriminative learning: Matrix and tensor framework". In: *arXiv preprint arXiv:1412.2863* (2014) (pp. 86, 87, 205).

[206] T. Jiang and N. Sidiropoulos. "A direct blind receiver for SIMO and MIMO OFDM systems subject to unknown frequency offset and multipath". In: *Proceedings of the 4th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. 2003, pp. 358–362 (p. 31).

[207] T. Jiang, N. Sidiropoulos, and J. M. ten Berge. "Almost-sure identifiability of multidimensional harmonic retrieval". In: *IEEE Transactions on Signal Processing* 49.9 (2001), pp. 1849–1859 (pp. 45, 170).

[208] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002 (pp. 3, 74).

[209] T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. J. Mckeown, V. Iragui, and T. J. Sejnowski. "Removing electroencephalographic artifacts by blind source separation". In: *Psychophysiology* 37.2 (2000), pp. 163–178 (p. 4).

[210] C. Jutten and J. Herault. “Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture”. In: *Signal Processing* 24.1 (1991), pp. 1–10 (pp. 6, 71, 94).

[211] E. Karahan, P. A. Rojas–Lopez, M. L. Bringas–Vega, P. A. Valdes–Hernandez, and P. A. Valdes–Sosa. “Tensor analysis and fusion of multimodal brain images”. In: *Proceedings of the IEEE* 103.9 (2015), pp. 1531–1559 (p. 30).

[212] E. Karvounis, M. Tsipouras, D. Fotiadis, and K. Naka. “An automated methodology for fetal heart rate extraction from the abdominal electrocardiogram”. In: *IEEE Transactions on Information Technology in Biomedicine* 11.6 (Nov. 2007), pp. 628–638 (p. 114).

[213] F. Kenney and E. Keeping. *Mathematics of statistics – Part Two*. D. Van Nostrand Company, Inc., 1951 (p. 83).

[214] B. Khoromskij. “Tensors-structured numerical methods in scientific computing: Survey on recent advances”. In: *Chemometrics and Intelligent Laboratory Systems* 110.1 (2012), pp. 1–19 (pp. 21, 29, 51, 52).

[215] B. N. Khoromskij. “$\mathcal{O}(d \log N)$-quantics approximation of n-d tensors in high-dimensional numerical modeling”. In: *Constructive Approximation* 34.2 (2011), pp. 257–280 (pp. 48, 52, 123, 127).

[216] H. A. Kiers. “A three-step algorithm for CANDECOMP/PARAFAC analysis of large data sets with multicollinearity”. In: *Journal of Chemometrics* 12.3 (1998), pp. 155–171 (p. 113).

[217] H. Kim and H. Park. “Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method”. In: *SIAM Journal on Matrix Analysis and Applications* 30.2 (2008), pp. 713–730 (p. 188).

[218] H. Kim and H. Park. “Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis”. In: *Bioinformatics* 23.12 (2007), pp. 1495–1502 (p. 188).

[219] T.-K. Kim, S.-F. Wong, and R. Cipolla. “Tensor canonical correlation analysis for action classification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007, pp. 1–8 (p. 30).

[220] J. G. Kirkwood. “Quantum statistics of almost classical assemblies”. In: *Physical Review* 44.1 (1933), p. 31 (p. 63).

[221] W. Koenig, H. Dunn, and L. Lacy. “The sound spectrograph”. In: *The Journal of the Acoustical Society of America* 18.1 (1946), pp. 19–49 (p. 61).

[222] E. Kofidis and P. A. Regalia. “On the best rank-1 approximation of higher-order supersymmetric tensors”. In: *SIAM Journal on Matrix Analysis and Applications* 23.3 (2002), pp. 863–884 (p. 40).

[223] T. G. Kolda and B. W. Bader. “Tensor decompositions and applications”. In: *SIAM Review* 51.3 (2009), pp. 455–500 (pp. 8, 20, 25, 95, 120, 149).

[224] T. G. Kolda, B. W. Bader, and J. P. Kenny. “Higher-order web link analysis using multilinear algebra”. In: *Proceedings of the 5th IEEE International Conference on Data Mining*. 2005 (p. 85).

[225] Y. Koren, R. Bell, C. Volinsky, et al. “Matrix factorization techniques for recommender systems”. In: *Computer* 42.8 (2009), pp. 30–37 (p. 188).

[226] T. Krazit. *How Amazon Web Services uses machine learning to make capacity planning decisions*. May 18, 2017. URL: `https://www.geekwire.com/2017/amazon-web-services-uses-machine-learning-make-capacity-planning-decisions/` (p. 2).

[227] W. Krijnen, T. Dijkstra, and A. Stegeman. “On the non-existence of optimal solutions and the occurrence of “degeneracy” in the CANDECOMP/PARAFAC model”. In: *Psychometrika* 73.3 (2008), pp. 431–439 (p. 133).

[228] H. Krim and M. Viberg. “Two decades of array signal processing: The parametric approach”. In: *IEEE Signal Processing Magazine* 13.4 (1996), pp. 67–94 (pp. 130, 141, 143).

[229] P. Kroonenberg. *Applied multiway data analysis*. Vol. 702. Wiley-Interscience, 2008 (pp. 8, 20).

[230] J. B. Kruskal. “Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics”. In: *Linear Algebra and its Applications* 18.2 (1977), pp. 95–138 (pp. 8, 20, 26, 76, 120, 149).

[231] S.-Y. Kung, K. S. Arun, and D. B. Rao. “State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem”. In: *Journal of the Optical Society of America* 73.12 (1983), pp. 1799–1811 (pp. 46, 49).

[232] D. Lahat, T. Adalı, and C. Jutten. “Multimodal data fusion: an overview of methods, challenges, and prospects”. In: *Proceedings of the IEEE* 103.9 (Sept. 2015), pp. 1449–1477 (pp. 28, 148).

[233] J. Landsberg. *Tensors: Geometry and applications*. Vol. 128. American Mathematical Society Providence, RI, USA, 2012 (pp. 8, 39, 80, 88).

[234] J. Landsberg. “The border rank of the multiplication of 2×2 matrices is seven”. In: *Journal of the American Mathematical Society* 19.2 (2006), pp. 447–459 (p. 39).

[235] A. N. Langville, C. D. Meyer, R. Albright, J. Cox, and D. Duling. "Initializations for the nonnegative matrix factorization". In: *Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. Citeseer. 2006, pp. 23–26 (p. 188).

[236] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta. "Massive MIMO for next generation wireless systems". In: *IEEE Communications Magazine* 52.2 (Feb. 2014), pp. 186–195 (pp. 119, 130, 141).

[237] J.-B. Lasserre. *Moments, positive polynomials and their applications*. Vol. 1. World Scientific, 2009 (pp. 17, 189, 190).

[238] M. Latka, Z. Was, A. Kozik, and B. J. West. "Wavelet analysis of epileptic spikes". In: *Physical Review E* 67.5 (2003), p. 052902 (p. 65).

[239] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. Vol. 15. SIAM, 1995 (p. 188).

[240] D. Lee, H. Seung, et al. "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755 (1999), pp. 788–791 (p. 188).

[241] S. E. Leurgans and R. T. Ross. "Multilinear models: Applications in spectroscopy". In: *Statistical Science* (1992), pp. 289–310 (p. 112).

[242] S. E. Leurgans, R. T. Ross, and R. B. Abel. "A decomposition for three-way arrays". In: *SIAM Journal on Matrix Analysis and Applications* 14.4 (1993), pp. 1064–1083 (pp. 29, 154).

[243] T. Li, T. Sauer, and J. Yorke. "The cheater's homotopy: An efficient procedure for solving systems of polynomial equations". In: *SIAM Journal on Numerical Analysis* 26.5 (1989), pp. 1241–1251 (p. 43).

[244] T.-H. Li and K. Mbarek. "A blind equalizer for nonstationary discrete-valued signals". In: *IEEE Transactions on Signal Processing* 45.1 (1997), pp. 247–254 (p. 147).

[245] X.-L. Li and X.-D. Zhang. "A family of generalized constant modulus algorithms for blind equalization". In: *IEEE Transactions on Communications* 54.11 (2006), pp. 1913–1917 (p. 146).

[246] Y. Li, A. Cichocki, and S.-I. Amari. "Analysis of sparse representation and blind source separation". In: *Neural computation* 16.6 (2004), pp. 1193–1234 (pp. 94, 118).

[247] J. Liang and Z. Ding. "Blind MIMO system identification based on cumulant subspace decomposition". In: *IEEE Transactions on Signal Processing* 51.6 (2003), pp. 1457–1468 (p. 89).

[248] L.-H. Lim and P. Comon. "Nonnegative approximations of nonnegative tensors". In: *Journal of Chemometrics* 23.7–8 (2009), pp. 432–441 (p. 133).

[249] L.-H. Lim and P. Comon. “Blind multilinear identification”. In: *Information Theory, IEEE Transactions on* 60.2 (2014), pp. 1260–1280 (p. 121).

[250] C.-J. Lin. “Projected gradient methods for nonnegative matrix factorization”. In: *Neural computation* 19.10 (2007), pp. 2756–2779 (pp. 188, 197).

[251] T. Lin and S. Bourennane. “Hyperspectral image processing by jointly filtering wavelet component tensor”. In: *IEEE Transactions on Geoscience and Remote Sensing* 51.6 (2013), pp. 3529–3541 (p. 65).

[252] Y. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. “MetaFac: Community discovery via relational hypergraph factorization”. In: *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 2009, pp. 527–536 (p. 85).

[253] H. Liu and G. Xu. “Closed-form blind symbol estimation in digital communications”. In: *IEEE Transactions on Signal Processing* 43.11 (1995), pp. 2714–2723 (p. 163).

[254] H. Liu and G. Xu. “Smart antennas in wireless systems: uplink multiuser blind channel and sequence detection”. In: *IEEE Transactions on Communications* 45.2 (1997), pp. 187–199 (pp. 162, 163).

[255] J. Liu, P. Musialski, P. Wonka, and J. Ye. “Tensor completion for estimating missing values in visual data”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (2013), pp. 208–220 (p. 30).

[256] Z. Liu, Y. Xin, and G. B. Giannakis. “Space–time–frequency-coded OFDM over frequency-selective fading channels”. In: *IEEE Transactions on Signal Processing* 50.10 (2002), pp. 2465–2476 (p. 61).

[257] L. Lócsi. “Approximating poles of complex rational functions”. In: *Acta Univ. Sapientiae Math* 1 (2009), pp. 169–182 (p. 114).

[258] L. Lócsi. “Rational function systems with applications in signal processing”. PhD thesis. Eötvös Loránd University, 2014 (p. 114).

[259] K. Löwner. “Über monotone Matrixfunktionen”. In: *Mathematische Zeitschrift* 38.1 (1934), pp. 177–216 (pp. 52, 53, 99).

[260] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. “A survey of multilinear subspace learning for tensor data”. In: *Pattern Recognition* 44.7 (2011), pp. 1540–1551 (p. 174).

[261] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. “MPCA: Multilinear principal component analysis of tensor objects”. In: *IEEE Transactions on Neural Networks* 19.1 (2008), pp. 18–39 (p. 30).

[262] W.-K. Ma, J. M. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C.-Y. Chi. "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing". In: *IEEE Signal Processing Magazine* 31.1 (2014), pp. 67–81 (p. 30).

[263] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. "Online learning for matrix factorization and sparse coding". In: *Journal of Machine Learning Research* 11.Jan (2010), pp. 19–60 (p. 61).

[264] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. "Supervised dictionary learning". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 22. 2009, pp. 1033–1040 (p. 6).

[265] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999 (pp. 21, 63).

[266] P. McCullagh. *Tensor methods in statistics*. Vol. 161. Chapman and Hall London, 1987 (pp. 79, 83).

[267] J. Mendel. "Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications". In: *Proceedings of the IEEE* 79.3 (Mar. 1991), pp. 278–305 (pp. 83, 94).

[268] S. Miron, Y. Song, D. Brie, and K. Wong. "Multilinear direction finding for sensor-array with multiple scales of invariance". In: *IEEE Transactions on Aerospace and Electronic Systems* 51.3 (July 2015), pp. 2057–2070 (pp. 130, 141).

[269] F. Miwakeichi, E. Martınez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi. "Decomposing EEG data into space–time–frequency components using parallel factor analysis". In: *NeuroImage* 22.3 (2004), pp. 1035–1045 (pp. 62, 65).

[270] M. J. Mohlenkamp and L. Monzón. "Trigonometric identities and sums of separable functions". In: *The Mathematical Intelligencer* 27.2 (2005), pp. 65–69 (p. 32).

[271] G. Moore. *Chrossing the chasm: Marketing and selling high-tech products to mainstream customers*. 3rd ed. Harper Collins, 2014 (p. 11).

[272] L. Mordell. "A new Waring's problem with squares of linear forms". In: *The Quarterly Journal of Mathematics* 1 (1930), pp. 276–288 (p. 40).

[273] M. Mørup, L. K. Hansen, and S. M. Arnfred. "ERPWAVELAB: A toolbox for multi-channel analysis of time–frequency transformed event related potentials". In: *Journal of Neuroscience Methods* 161.2 (2007), pp. 361–368 (p. 62).

[274] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred. "Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG". In: *NeuroImage* 29.3 (2006), pp. 938–947 (p. 65).

[275] S. P. Neugebauer. “A deterministic blind equalization method for multi-modulus signals”. In: *Proceedings of the Fourth IEEE Workshop on Sensor Array and Multichannel Signal Processing*. 2006, pp. 551–555 (pp. 147, 150).

[276] C. L. Nikias and A. P. Petropulu. *Higher-order spectra analysis: A nonlinear signal processing framework*. PTR Prentice Hall, Englewood Cliffs, NJ, 1993 (p. 78).

[277] C. L. Nikias and J. M. Mendel. “Signal processing with higher-order spectra”. In: *IEEE Signal Processing Magazine* 10.3 (1993), pp. 10–37 (p. 79).

[278] J. Nocedal and S. J. Wright. *Numerical optimization*. 2nd ed. Springer, 2006 (p. 192).

[279] M. Novey and T. Adalı. “Complex fixed-point ICA algorithm for separation of QAM sources using Gaussian mixture model”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 2. 2007, pp. 445–448 (p. 164).

[280] M. F. Ochs, R. Stoyanova, F. Arias-Mendoza, and T. Brown. “A new method for spectral decomposition using a bilinear Bayesian approach”. In: *Journal of Magnetic Resonance* 137.1 (1999), pp. 161–176 (p. 196).

[281] L. Oeding and G. Ottaviani. “Eigenvectors of tensors and algorithms for Waring decomposition”. In: *Journal of Symbolic Computation* 54 (2013), pp. 9–35 (p. 40).

[282] K. N. Oh and Y. O. Chin. “New blind equalization techniques based on constant modulus algorithm”. In: *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*. Vol. 2. Nov. 1995, pp. 865–869 (pp. 146, 147, 169).

[283] R. Orús. “A practical introduction to tensor networks: Matrix product states and projected entangled pair states”. In: *Annals of Physics* 349 (2014), pp. 117–158 (p. 28).

[284] M. Osborne. “Some special nonlinear least squares problems”. In: *SIAM Journal on Numerical Analysis* 12.4 (1975), pp. 571–592 (p. 49).

[285] I. V. Oseledets. “Approximation of $2^d \times 2^d$ matrices using tensor decomposition”. In: *SIAM Journal on Matrix Analysis and Applications* 31.4 (2010), pp. 2130–2145 (p. 52).

[286] I. V. Oseledets. “Tensor-train decomposition”. In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317 (pp. 9, 28, 118, 185).

[287] I. Ovejero-Lopez, R. Bro, and W. Bredie. “Univariate and multivariate modelling of flavour release in chewing gum using time-intensity: A comparison of data analytical methods”. In: *Food Quality and Preference* 16.4 (2005), pp. 327–343 (p. 30).

[288] A. Ozerov and C. Févotte. “Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.3 (2010), pp. 550–563 (p. 62).

[289] P. Paatero and U. Tapper. “Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values”. In: *Environmetrics* 5.2 (1994), pp. 111–126 (p. 188).

[290] J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens, and R. Pintelon. “Identification of nonlinear systems using polynomial nonlinear state space models”. In: *Automatica* 46.4 (2010), pp. 647–656 (p. 36).

[291] L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank citation ranking: Bringing order to the web.* Technical Report. 1999 (p. 3).

[292] E. E. Papalexakis, L. Akoglu, and D. Ience. “Do more views of a graph help? Community detection and clustering in multi-graphs”. In: *Proceedings of the 16th IEEE International Conference on Information fusion (FUSION).* 2013, pp. 899–905 (p. 85).

[293] J.-M. Papy. “Subspace-based exponential data fitting using linear and multilinear data”. PhD thesis. KU Leuven, 2005 (pp. 12, 14, 49).

[294] J.-M. Papy, L. De Lathauwer, and S. Van Huffel. “Exponential data fitting using multilinear algebra: The single-channel and multi-channel case”. In: *Numerical Linear Algebra with Applications* 12.8 (2005), pp. 809–826 (pp. 48, 49).

[295] D.-T. Pham and J.-F. Cardoso. “Blind separation of instantaneous mixtures of nonstationary sources”. In: *IEEE Transactions on Signal Processing* 49.9 (2001), pp. 1837–1848 (p. 72).

[296] A.-H. Phan, P. Tichavský, and A. Cichocki. “Blind source separation of single channel mixture using tensorization and tensor diagonalization”. In: *Latent Variable Analysis and Signal Separation.* Vol. 10169. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2017, pp. 36–46 (p. 45).

[297] A.-H. Phan, P. Tichavský, and A. Cichocki. “Fast alternating LS algorithms for high order CANDECOMP/PARAFAC tensor factorizations”. In: *IEEE Transactions on Signal Processing* 61.19 (2013), pp. 4834–4846 (p. 29).

[298] A. Plaza, P. Martinez, R. Perez, and J. Plaza. "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data". In: *IEEE Transactions on Geoscience and Remote Sensing* 42.3 (Mar. 2004), pp. 650–663 (p. 194).

[299] G. Pólya and G. Szegö. *Problems and Theorems in Analysis II: Theory of Functions*. Springer Science & Business Media, 1997 (p. 190).

[300] V. Powers and B. Reznick. "Polynomials that are positive on an interval". In: *Transactions of the American Mathematical Society* 352.10 (2000), pp. 4677–4692 (p. 190).

[301] L. Qi. "Eigenvalues of a real supersymmetric tensor". In: *Journal of Symbolic Computation* 40.6 (2005), pp. 1302–1324 (p. 40).

[302] L. Qi and Z. Luo. *Tensor analysis: Spectral theory and special tensors*. SIAM, 2017 (pp. 40, 66).

[303] L. Qi, F. Wang, and Y. Wang. "Z-eigenvalue methods for a global polynomial optimization problem". In: *Mathematical Programming* 118.2 (2009), pp. 301–316 (p. 40).

[304] S. Rendle. "Factorization machines". In: *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM)*. 2010, pp. 995–1000 (p. 40).

[305] S. Rickard, R. Balan, and J. Rosca. "Real-time time–frequency based blind source separation". In: *Proceedings of the third International Conference on Independent Component Analysis and Signal Separation (ICA)*. 2001 (pp. 61, 72).

[306] A. Rihaczek. "Signal energy distribution in time and frequency". In: *IEEE Transactions on Information Theory* 14.3 (1968), pp. 369–374 (p. 63).

[307] F. Roemer, M. Haardt, and G. Del Galdo. "Analytical performance assessment of multi-dimensional matrix- and tensor-based ESPRIT-type algorithms". In: *IEEE Transactions on Signal Processing* 62.10 (2014), pp. 2611–2625 (p. 49).

[308] L. Rota and P. Comon. "Blind equalizers based on polynomial criteria". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 4. 2004, pp. 441–444 (p. 147).

[309] R. Roy and T. Kailath. "ESPRIT-estimation of signal parameters via rotational invariance techniques". In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 37.7 (1989), pp. 984–995 (pp. 45, 49).

[310] B. Rubehn, C. Bosman, R. Oostenveld, P. Fries, and T. Stiegliz. “A MEMS-based flexible multichannel ECoG-electrode array”. In: *Journal of Neural Engineering* 6.3 (2009), pp. 1–10 (p. 130).

[311] P. Sajda, S. Du, T. R. Brown, R. Stoyanova, D. C. Shungu, X. Mao, and L. C. Parra. “Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain”. In: *IEEE Transactions on Medical Imaging* 23.12 (2004), pp. 1453–1465 (p. 196).

[312] E. Sanchez and B. R. Kowalski. “Tensorial resolution: A direct trilinear decomposition”. In: *Journal of Chemometrics* 4.1 (1990), pp. 29–45 (p. 29).

[313] M. N. Schmidt and H. Laurberg. “Nonnegative matrix factorization with Gaussian process priors”. In: *Computational Intelligence and Neuroscience* (2008) (pp. 194, 196).

[314] D. Seo, J. M. Carmena, J. M. Rabaey, E. Alon, and M. M. Maharbiz. “Neural dust: An ultrasonic, low power solution for chronic brain–machine interfaces”. In: *arXiv preprint arXiv:1307.2196* (2013) (p. 130).

[315] W. Sethares, G. Rey, C. R. Johnson Jr, et al. “Approaches to blind equalization of signals with multiple modulus”. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 1989, pp. 972–975 (p. 147).

[316] S. A. W. Shah, K. Abed-Meraim, and T. Y. Al-Naffouri. “Multi-modulus algorithms using hyperbolic and Givens rotations for blind deconvolution of MIMO systems”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 2155–2159 (pp. 16, 147, 164).

[317] S. A. W. Shah, K. Abed-Meraim, and T. Y. Al-Naffouri. “Multi-modulus algorithms using hyperbolic and Givens rotations for MIMO deconvolution”. In: *arXiv preprint arXiv:1506.06650* (2015) (pp. 16, 164).

[318] T.-J. Shan, M. Wax, and T. Kailath. “On spatial smoothing for direction-of-arrival estimation of coherent signals”. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 33.4 (Aug. 1985), pp. 806–811 (p. 142).

[319] E. Al-Sharoa, M. Al-khassaweneh, and S. Aviyente. “A tensor-based framework for community detection in dynamic networks”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2017, pp. 2312–2316 (p. 85).

[320] Y. Shen, B. Baingana, and G. B. Giannakis. "Inferring directed network topologies via tensor factorization". In: *Proceedings of the 50th Asilomar Conference on Signals, Systems and Computers*. IEEE. Nov. 2016, pp. 1739–1743 (pp. 72, 85).

[321] N. D. Sidiropoulos, E. E. Papalexakis, and C. Faloutsos. "Parallel randomly compressed cubes: A scalable distributed architecture for big tensor decomposition". In: *IEEE Signal Processing Magazine* 31.5 (Sept. 2014), pp. 57–70 (pp. 133, 144).

[322] N. Sidiropoulos and R. Bro. "On the uniqueness of multilinear decomposition of N-way arrays". In: *Journal of Chemometrics* 14.3 (2000), pp. 229–239 (pp. 8, 26, 45, 121, 141).

[323] N. Sidiropoulos, R. Bro, and G. Giannakis. "Parallel factor analysis in sensor array processing". In: *IEEE Transactions on Signal Processing* 48.8 (2000), pp. 2377–2388 (pp. 6, 12, 14, 30, 45, 49, 56, 130, 146).

[324] N. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. Papalexakis, and C. Faloutsos. "Tensor decomposition for signal processing and machine learning". In: *IEEE Transactions on Signal Processing* 65.13 (July 2017), pp. 3551–3582 (pp. 8, 20, 25, 26, 82, 120, 121, 149, 174).

[325] N. Sidiropoulos, G. Giannakis, and R. Bro. "Blind PARAFAC receivers for DS-CDMA systems". In: *IEEE Transactions on Signal Processing* 48.3 (2000), pp. 810–823 (pp. 31, 50).

[326] N. Sidiropoulos and A. Kyrillidis. "Multi-way compressed sensing for sparse low-rank tensors". In: *IEEE Signal Processing Letters* 19.11 (2012), pp. 757–760 (p. 27).

[327] D. Silva and L.-H. Lim. "Tensor rank and the ill-posedness of the best low-rank approximation problem". In: *SIAM Journal on Matrix Analysis and Applications* 30.3 (2008), pp. 1084–1127 (p. 133).

[328] A. K. Smilde, R. Bro, P. Geladi, and J. Wiley. *Multi-way analysis with applications in the chemical sciences*. Wiley Chichester, UK, 2004 (pp. 20, 30, 113, 174).

[329] M. Smith and T. Barnwell. "Exact reconstruction techniques for tree-structured subband coders". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34.3 (1986), pp. 434–441 (p. 66).

[330] Y. Song and L. Qi. "Infinite and finite dimensional Hilbert tensors". In: *Linear Algebra and Its Applications* 451 (2014), pp. 1–14 (p. 48).

[331] L. Sorber, M. Van Barel, and L. De Lathauwer. *Complex optimization toolbox v1.0*. Available online. `http://esat.kuleuven.be/stadius/cot/`. Feb. 2013 (pp. 174, 194).

[332] L. Sorber, M. Van Barel, and L. De Lathauwer. "Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization". In: *SIAM Journal on Optimization* 23.2 (2013), pp. 695–720 (pp. 29, 109, 129, 133, 175, 181).

[333] L. Sorber, M. Van Barel, and L. De Lathauwer. "Structured data fusion". In: *IEEE Journal of Selected Topics in Signal Processing* 9.4 (June 2015), pp. 586–600 (pp. 28, 109, 133, 148, 155, 175, 178, 179, 184, 197, 206).

[334] L. Sorber, M. Van Barel, and L. De Lathauwer. *Tensorlab v2.0*. Available online. `http://www.tensorlab.net`. Jan. 2014 (p. 109).

[335] L. Sorber, M. Van Barel, and L. De Lathauwer. "Unconstrained optimization of real functions in complex variables". In: *SIAM Journal on Optimization* 22.3 (2012), pp. 879–898 (pp. 174, 192, 194).

[336] L. Sorber, M. Van Barel, and L. De Lathauwer. "Numerical solution of bivariate and polyanalytic polynomial systems". In: *SIAM Journal on Numerical Analysis* 52.4 (2014), pp. 1551–1572 (pp. 43, 44).

[337] M. Sørensen and L. De Lathauwer. "Blind signal separation via tensor decomposition with Vandermonde factor: Canonical polyadic decomposition". In: *IEEE Transactions on Signal Processing* 61.22 (Nov. 2013), pp. 5507–5519 (pp. 28, 49, 130).

[338] M. Sørensen and L. De Lathauwer. "Convolutive low-rank factorizations via coupled low-rank and Toeplitz structured matrix/tensor decompositions". Technical Report 16–37, ESAT–STADIUS, KU Leuven, Belgium. Mar. 2016 (p. 163).

[339] M. Sørensen and L. De Lathauwer. "Coupled tensor decompositions and monomial structure in array processing". Technical Report 15–152, ESAT–STADIUS, KU Leuven, Belgium. 2015 (pp. 14, 57, 59, 60, 78).

[340] M. Sørensen and L. De Lathauwer. "Multidimensional harmonic retrieval via coupled canonical polyadic decomposition — Part I: Model and identifiability". In: *IEEE Transactions on Signal Processing* 65.2 (Jan. 2017), pp. 517–527 (pp. 28, 49).

[341] M. Sørensen and L. De Lathauwer. "Multidimensional harmonic retrieval via coupled canonical polyadic decomposition — Part II: Algorithm and multirate sampling". In: *IEEE Transactions on Signal Processing* 65.2 (Jan. 2017), pp. 528–539 (p. 49).

[342] M. Sørensen and L. De Lathauwer. "Multiple invariance ESPRIT for nonuniform linear arrays: A coupled canonical polyadic decomposition approach". In: *IEEE Transactions on Signal Processing* 64.14 (July 2016), pp. 3693–3704 (p. 148).

[343] M. Sørensen and L. De Lathauwer. "Shift invariance, incomplete arrays and coupled CPD: A case study". In: *Proceedings of the 9th IEEE Sensor Array and Multichannel Signal Processing Workshop*. 2016 (p. 148).

[344] M. Sørensen, L. Lathauwer, P. Comon, S. Icart, and L. Deneire. "Canonical polyadic decomposition with a columnwise orthonormal factor matrix". In: *SIAM Journal on Matrix Analysis and Applications* 33.4 (2012), pp. 1190–1213 (pp. 133, 170, 172).

[345] M. Sørensen and L. De Lathauwer. "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms — Part I: Uniqueness". In: *SIAM Journal on Matrix Analysis and Applications* 36.2 (2015), pp. 496–522 (pp. 9, 26, 28, 60, 107, 121, 148, 155, 160, 206).

[346] M. Sørensen, I. Domanov, and L. De Lathauwer. "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms — Part II: Algorithms". In: *SIAM Journal on Matrix Analysis and Applications* 36.3 (2015), pp. 1015–1045 (pp. 9, 28, 77, 107, 148, 155, 160, 206).

[347] A. Souloumiac. "Blind source detection and separation using second order non-stationarity". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 3. 1995, pp. 1912–1915 (p. 72).

[348] A. Stegeman. "On uniqueness of the nth order tensor decomposition into rank-1 terms with linear independence in one mode". In: *SIAM Journal on Matrix Analysis and Applications* 31.5 (2010), pp. 2498–2516 (p. 26).

[349] G. Strang and T. Nguyen. *Wavelets and filter banks*. SIAM, 1996 (p. 66).

[350] V. Strassen. "Gaussian elimination is not optimal". In: *Numerische Mathematik* 13.4 (Aug. 1969), pp. 354–356 (pp. 38, 39).

[351] J. Sun, D. Tao, and C. Faloutsos. "Beyond streams and graphs: Dynamic tensor analysis". In: *Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data mining (SIGKDD)*. 2006, pp. 374–383 (p. 7).

[352] S. Talwar, M. Viberg, and A. Paulraj. "Blind estimation of multiple co-channel digital signals arriving at an antenna array". In: *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*. IEEE. 1993, pp. 349–353 (p. 153).

[353] The Economist. *Data is giving rise to a new economy: Fuel of the future*. May 6, 2017. URL: `http://www.economist.com/news/briefing/21721634-how-it-shaping-up-data-giving-rise-new-economy` (p. 2).

[354] T. N. Thiele. “The theory of observations”. In: *The Annals of Mathematical Statistics* 2 (1931), pp. 165–308 (p. 82).

[355] E. G. Thomas. “A polarization identity for multilinear maps”. In: *Indagationes Mathematicae* 25.3 (2014), pp. 468–474 (pp. 80, 88).

[356] K. Thulasiraman and M. N. Swamy. *Graphs: Theory and algorithms*. John Wiley & Sons, 2011 (p. 84).

[357] P. Tichavský and A. Yeredor. “Fast approximate joint diagonalization incorporating weight matrices”. In: *IEEE Transactions on Signal Processing* 57.3 (2009), pp. 878–891 (p. 84).

[358] F. Tisseur and K. Meerbergen. “The quadratic eigenvalue problem”. In: *SIAM review* 43.2 (2001), pp. 235–286 (p. 150).

[359] G. Tomasi and R. Bro. “PARAFAC and missing values”. In: *Chemometrics and Intelligent Laboratory Systems* 75.2 (2005), pp. 163–180 (p. 184).

[360] L. Tong, V. Soon, Y. Huang, and R. Liu. “AMUSE: A new blind identification algorithm”. In: *International Symposium on Circuits and Systems*. IEEE. 1990, pp. 1784–1787 (p. 72).

[361] J. Toonders. *Data is the new oil of the economy*. July 2014. URL: `https://www.wired.com/insights/2014/07/data-new-oil-digital-economy/` (p. 1).

[362] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997 (p. 123).

[363] J. Treichler and B. Agee. “A new approach to multipath correction of constant modulus signals”. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 31.2 (1983), pp. 459–472 (pp. 75, 146, 153).

[364] L. R. Tucker. “Some mathematical notes on three-mode factor analysis”. In: *Psychometrika* 31.3 (1966), pp. 279–311 (pp. 9, 20).

[365] D. W. Tufts and R. Kumaresan. “Estimation of frequencies of multiple sinusoids: Making linear prediction perform like maximum likelihood”. In: *Proceedings of the IEEE* 70.9 (1982), pp. 975–989 (p. 49).

[366] M. A. Turk and A. P. Pentland. “Face recognition using eigenfaces”. In: *Proceedings of the IEEE Computer Science Conference on Computer Vision and Pattern Recognition (CVPR)*. 1991, pp. 586–591 (pp. 3, 8, 30).

[367] V. Turner. *The Internet of Things: Getting ready to embrace its impact on the digital economy*. IDC Directions, Mar. 2016 (p. 1).

[368] E. Tyrtyshnikov. “How bad are Hankel matrices?” In: *Numerische Mathematik* 67.2 (1994), pp. 261–269 (p. 124).

[369] United Nations. *World population prospects: The 2015 revision*. 2015. URL: https://esa.un.org/unpd/wpp/ (p. 1).

[370] A. Uschmajew. "Local convergence of the alternating least squares algorithm for canonical tensor approximation". In: *SIAM Journal on Matrix Analysis and Applications* 33.2 (2012), pp. 639–652 (p. 175).

[371] P. P. Vaidyanathan. "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial". In: *Proceedings of the IEEE* 78.1 (1990), pp. 56–93 (pp. 66, 162).

[372] M. H. Van Benthem and M. R. Keenan. "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems". In: *Journal of Chemometrics* 18.10 (2004), pp. 441–450 (p. 188).

[373] F. Van Eeghem, M. Sørensen, and L. De Lathauwer. "Tensor decompositions with several block-Hankel factors and application in blind system identification". In: *IEEE Transactions on Signal Processing* (Apr. 2017). To appear (p. 89).

[374] M. Vandecappelle, M. Boussé, F. Van Eeghem, and L. De Lathauwer. "Tensor decompositions for graph clustering". Technical Report 16–170, ESAT–STADIUS, KU Leuven, Belgium. 2016 (pp. 85, 86).

[375] J. Vanderstukken, A. Stegeman, and L. De Lathauwer. "Sets of polynomial equations, decompositions of higher-order tensors and multidimensional harmonic retrieval: Connections and algorithms". MA thesis. KU Leuven, 2017 (p. 43).

[376] D. Vandevoorde. "A fast exponential decomposition algorithm and its applications to structured matrices". PhD thesis. Troy, NY: Rensselaer Polytechnic Institute, 1998 (pp. 46, 48, 101).

[377] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen. "A new truncation strategy for the higher-order singular value decomposition". In: *SIAM Journal on Scientific Computing* 34.2 (2012), A1027–A1052 (p. 182).

[378] M. A. O. Vasilescu and D. Terzopoulos. "Multilinear (tensor) ICA and dimensionality reduction". In: *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA)*. Vol. 4666. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, pp. 818–826 (p. 30).

[379] M. A. O. Vasilescu and D. Terzopoulos. "Multilinear analysis of image ensembles: Tensorfaces". In: *Proceedings of the 7th European Conference on Computer Vision (ECCV)*. Ed. by A. Heyden, G. Sparr, M. Nielsen, and P. Johansen. Vol. 2350. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2002, pp. 447–460 (p. 30).

[380] S. A. Vavasis. “On the complexity of nonnegative matrix factorization”. In: *SIAM Journal on Optimization* 20.3 (2009), pp. 1364–1377 (p. 192).

[381] Z. Vavřín. “A unified approach to Löwner and Hankel matrices”. In: *Linear Algebra and its Applications* 143 (1991), pp. 171–222 (pp. 53, 54, 99, 108).

[382] Z. Vavřín. “Confluent Cauchy and Cauchy-Vandermonde matrices”. In: *Linear Algebra and its Applications* 258 (1997), pp. 271–293 (pp. 53, 54, 102, 104, 108).

[383] A.-J. van der Veen. “Analytical method for blind binary signal separation”. In: *IEEE Transactions on Signal Processing* 45.4 (1997), pp. 1078–1082 (p. 77).

[384] A.-J. van der Veen and A. Paulraj. “An analytical constant modulus algorithm”. In: *IEEE Transactions on Signal Processing* 44.5 (1996), pp. 1136–1155 (pp. 13, 20, 27, 74, 75, 146, 153, 164, 178, 202).

[385] A.-J. van der Veen and J. Tol. “Separation of zero/constant modulus signals”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 5. 1997, pp. 3445–3448 (pp. 77, 169).

[386] A.-J. van der Veen, S. Talwar, and A. Paulraj. “A subspace approach to blind space-time signal processing for wireless communication systems”. In: *IEEE Transactions on Signal Processing* 45.1 (1997), pp. 173–190 (p. 162).

[387] A.-J. van der Veen and A. Trindade. “Combining blind equalization with constant modulus properties”. In: *Proceedings of the 34th Asilomar Conference on Signals, Systems and Computers*. Vol. 2. IEEE. 2000, pp. 1568–1572 (p. 146).

[388] N. Vervliet, O. Debals, and L. De Lathauwer. “Canonical polyadic decomposition of incomplete tensors with linearly constrained factors”. Technical Report 16–172, ESAT–STADIUS, KU Leuven, Belgium. Apr. 2017 (pp. 17, 184).

[389] N. Vervliet, O. Debals, and L. De Lathauwer. “Exploiting efficient data representations in large-scale tensor decompositions”. Technical Report 16–174, ESAT–STADIUS, KU Leuven, Belgium. Aug. 2017 (pp. 49, 54, 185).

[390] N. Vervliet, O. Debals, and L. De Lathauwer. “Tensorlab 3.0 — Numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization”. In: *Proceedings of the 50th Asilomar Conference on Signals, Systems and Computers*. IEEE. Nov. 2016, pp. 1733–1738 (pp. 49, 54, 173).

[391] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer. "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis". In: *IEEE Signal Processing Magazine* 31.5 (Sept. 2014), pp. 71–79 (pp. 14, 17, 29, 49, 97, 118, 133, 144, 178, 183, 184).

[392] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer. *Tensorlab 3.0*. Mar. 2016. URL: `http://www.tensorlab.net` (pp. 29, 133, 155, 164, 174, 197, 204).

[393] N. Vervliet and L. De Lathauwer. "A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors". In: *IEEE Journal of Selected Topics in Signal Processing* 10.2 (2016), pp. 284–295 (pp. 17, 29, 133, 144, 184).

[394] M. Vetterli and C. Herley. "Wavelets and filter banks: Theory and design". In: *IEEE Transactions on Signal Processing* 40.9 (1992), pp. 2207–2232 (p. 63).

[395] J. Ville et al. "Théorie et applications de la notion de signal analytique". In: *Cables et transmission* 2.1 (1948), pp. 61–74 (p. 62).

[396] Y. Wang, Y. Pati, Y. Cho, A. Paulraj, and T. Kailath. "A matrix factorization approach to signal copy of constant modulus signals arriving at an antenna array". In: *Proc. 28th Conference on Information Sciences and Systems*. 1994, pp. 178–183 (p. 153).

[397] M. Weis, F. Römer, M. Haardt, D. Jannek, and P. Husar. "Multi-dimensional space–time–frequency component analysis of event related EEG data using closed-form PARAFAC". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2009, pp. 349–352 (p. 65).

[398] E. Wigner. "On the quantum correction for thermodynamic equilibrium". In: *Physical review* 40.5 (1932), p. 749 (p. 62).

[399] S. Wild, J. Curry, and A. Dougherty. "Improving non-negative matrix factorizations through structured initialization". In: *Pattern Recognition* 37.11 (2004), pp. 2217–2232 (p. 188).

[400] T. Wu, A. R. Benson, and D. F. Gleich. "General tensor spectral co-clustering for higher-order data". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 29. 2016, pp. 2559–2567 (p. 85).

[401] J. Yang. "Multimodulus algorithms for blind equalization". PhD thesis. University of British Columbia, 1997 (pp. 146, 170).

[402] J. Yang, J.-J. Werner, and G. Dumont. "The multimodulus blind equalization algorithm". In: *IEEE International Conference on Digital Signal Processing (DSP)*. Vol. 1. 1997, pp. 127–130 (pp. 146, 170).

[403] A. Yeredor. “Blind separation of Gaussian sources via second-order statistics with asymptotically optimal weighting”. In: *IEEE Signal Processing Letters* 7.7 (2000), pp. 197–200 (p. 84).

[404] A. Yeredor. “Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation”. In: *IEEE Transactions on Signal Processing* 50.7 (2002), pp. 1545–1553 (p. 84).

[405] A. Yeredor. “Blind source separation via the second characteristic function”. In: *Signal Processing* 80.5 (2000), pp. 897–902 (p. 73).

[406] O. Yilmaz and S. Rickard. “Blind separation of speech mixtures via time–frequency masking”. In: *IEEE Transactions on Signal Processing* 52.7 (2004), pp. 1830–1847 (pp. 61, 72).

[407] K. Y. Yılmaz, A. T. Cemgil, and U. Simsekli. “Generalised coupled tensor factorisation”. In: *Advances in Neural Information Processing Systems (NIPS)*. Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger. Vol. 24. Curran Associates, Inc., 2011, pp. 2151–2159 (p. 28).

[408] T. Yokota, R. Zdunek, A. Cichocki, and Y. Yamashita. “Smooth nonnegative matrix and tensor factorizations for robust multi-way data analysis”. In: *Signal Processing* 113 (2015), pp. 234–249 (pp. 94, 188, 191).

[409] T. Yokota, Q. Zhao, and A. Cichocki. “Smooth PARAFAC decomposition for tensor completion”. In: *IEEE Transactions on Signal Processing* 64.20 (2015), pp. 5423–5436 (p. 30).

[410] Y. Yu, J. Jin, F. Liu, and S. Crozier. “Multidimensional compressed sensing MRI using tensor decomposition-based sparsifying transform”. In: *PloS one* 9.6 (2014), pp. 1–13 (p. 29).

[411] V. Zarzoso and P. Comon. “Blind and semi-blind equalization based on the constant power criterion”. In: *IEEE Transactions on Signal Processing* 53.11 (2005), pp. 4363–4375 (pp. 77, 152).

[412] V. Zarzoso and P. Comon. “Robust independent component analysis by iterative maximization of the kurtosis contrast with algebraic optimal step size”. In: *IEEE Transactions on Neural Networks* 21.2 (2010), pp. 248–261 (p. 164).

[413] Q. Zhang, M. Berry, B. Lamb, and T. Samuel. “A parallel nonnegative tensor factorization algorithm for mining global climate data”. In: *Proceedings of the International Conference on Computational Science (ICCS)*. Springer, 2009, pp. 405–415 (p. 50).

[414] S. Zhou and G. Giannakis. “Finite-alphabet based channel estimation for OFDM and related multicarrier systems”. In: *IEEE Transactions on Communications* 49.8 (Aug. 2001), pp. 1402–1414 (p. 147).

[415] M. Zibulevsky and B. Pearlmutter. “Blind source separation by sparse decomposition in a signal dictionary”. In: *Neural computation* 13.4 (2001), pp. 863–882 (pp. 61, 94, 118).

Curriculum

Otto Debals was born on December 27, 1990 in Kortrijk, Belgium.

Degrees & Education

Sep 2013 – Sep 2017	PhD Candidate in Engineering, **KU Leuven** – Supported by a doctoral fellowship of IWT – Two-week research stay in National University of Athens *Topics: multilinear algebra, signal processing, ...*
Sep 2011 – Jul 2013	Master of Science in Engineering, **KU Leuven** Mathematical Engineering 88.31% *Summa cum laude – Best of class*
Sep 2008 – Jul 2011	Bachelor of Science in Engineering, **KU Leuven** Electrical Engineering / Computer Science 88.78% *Summa cum laude – Best of class*
Sep 2006 – Jul 2008	Mathematics & Natural Sciences, **Sint-Amandscollege**, Kortrijk
Sep 2002 – Jul 2006	Latin & Mathematics, **Sint-Amandscollege**, Kortrijk

Relevant Experience

Jun 2014	Passed **CFA** Level I *Chartered Financial Analyst Program*
Aug – Sep 2012	Internship at **McKinsey & Company**, Brussels *Summer Business Analyst — Energy and mining*
Summer 2009 – 2011	Three summer projects at **KULAK**, Kortrijk & Internship at **Barco NV**, Kortrijk *Developments, implementations and simulations*

List of publications

Articles in peer-reviewed journals

Published

[1] M. Boussé, O. Debals, L. De Lathauwer. "Tensor-based large-scale blind system identification using segmentation". In: *IEEE Transactions on Signal Processing* (Sept. 2017). To appear.

[2] O. Debals, M. Van Barel, L. De Lathauwer. "Nonnegative matrix factorization using nonnegative polynomial approximations". In: *IEEE Signal Processing Letters* 24.7 (July 2017), pp. 948–952.

[3] M. Boussé, O. Debals, L. De Lathauwer. "A tensor-based method for large-scale blind source separation using segmentation". In: *IEEE Transactions on Signal Processing* 65.2 (Jan. 2017), pp. 346–358.

[4] O. Debals, M. Van Barel, L. De Lathauwer. "Löwner-based blind signal separation of rational functions with applications". In: *IEEE Transactions on Signal Processing* 64.8 (Apr. 2016), pp. 1909–1918.

[5] N. Vervliet, O. Debals, L. Sorber, L. De Lathauwer. "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis". In: *IEEE Signal Processing Magazine* 31.5 (Sept. 2014), pp. 71–79.

To be submitted or under review

[1] O. Debals, L. De Lathauwer. "The concept of tensorization". Technical Report 17–99, ESAT–STADIUS, KU Leuven, Belgium. Aug. 2017.

[2] F. Van Eeghem, O. Debals, L. De Lathauwer. "Coupled and incomplete tensors in blind system identification". Technical Report 17–128, ESAT–STADIUS, KU Leuven, Belgium. Aug. 2017.

[3] N. Vervliet, O. Debals, L. De Lathauwer. "Exploiting efficient data representations in large-scale tensor decompositions". Technical Report 16–174, ESAT–STADIUS, KU Leuven, Belgium. Aug. 2017.

[4] F. Van Eeghem, O. Debals, L. De Lathauwer. "Tensor similarity in two modes". Technical Report 17–97, ESAT–STADIUS, KU Leuven, Belgium. May 2017.

[5] N. Vervliet, O. Debals, L. De Lathauwer. "Canonical polyadic decomposition of incomplete tensors with linearly constrained factors". Technical Report 16–172, ESAT–STADIUS, KU Leuven, Belgium. Apr. 2017.

[6] M. Boussé, N. Vervliet, I. Domanov, O. Debals, L. De Lathauwer. "Linear systems with a canonical polyadic decomposition constrained solution: Algorithms and applications". Technical Report 17–01, ESAT–STADIUS, KU Leuven, Belgium. Jan. 2017.

[7] H. Bharath, O. Debals, D. Sima, U. Himmelreich, L. De Lathauwer, S. Van Huffel. "Löwner-based method for residual water suppression in ^{1}H magnetic resonance spectroscopic imaging". Technical Report 17–73, ESAT–STADIUS, KU Leuven, Belgium. Jan. 2017.

[8] O. Debals, M. Sohail, L. De Lathauwer. "Analytical multi-modulus algorithms based on coupled canonical polyadic decompositions". Technical Report 16–150, ESAT–STADIUS, KU Leuven, Belgium. Sept. 2016.

Articles in peer-reviewed conference proceedings

Published

[1] M. Boussé, G. Goovaerts, N. Vervliet, O. Debals, S. Van Huffel, L. De Lathauwer. "Irregular heartbeat classification using Kronecker product equations". In: *Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. To appear. July 2017.

[2] I. Markovsky, O. Debals, L. De Lathauwer. "Sum-of-exponentials modeling and common dynamics estimation using Tensorlab". In: *Proceedings of the 20th World Congress of the International Federation of Automatic Control (IFAC)*. July 2017, pp. 14715–14720.

[3] N. Vervliet, O. Debals, L. De Lathauwer. "Tensorlab 3.0 — Numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization". In: *Proceedings of the 50th Asilomar Conference on Signals, Systems and Computers*. IEEE. Nov. 2016, pp. 1733–1738.

[4] M. Boussé, O. Debals, L. De Lathauwer. "A tensor-based method for large-scale blind system identification using segmentation". In: *Proceedings of the 24th IEEE European Signal Processing Conference (EUSIPCO)*. Aug. 2016, pp. 2015–2019.

[5] X. F. Gong, Q. H. Lin, O. Debals, N. Vervliet, L. De Lathauwer. "Coupled rank-$(L_m, L_n, \cdot)$ block term decomposition by coupled block simultaneous generalized Schur decomposition". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2016, pp. 2554–2558.

[6] O. Debals, L. De Lathauwer. "Stochastic and deterministic tensorization for blind signal separation". In: *Latent Variable Analysis and Signal Separation*. Vol. 9237. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, Aug. 2015, pp. 3–13.

[7] M. Boussé, O. Debals, L. De Lathauwer. "A novel deterministic method for large-scale blind source separation". In: *Proceedings of the 23rd IEEE European Signal Processing Conference (EUSIPCO)*. Aug. 2015, pp. 1890–1894.

[8] O. Debals, M. Van Barel, L. De Lathauwer. "Blind signal separation of rational functions using Löwner-based tensorization". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2015, pp. 4145–4149.

Under review

[1] M. Boussé, N. Vervliet, O. Debals, L. De Lathauwer. "Face recognition as a Kronecker product equation". Technical Report 17–41, ESAT–STADIUS, KU Leuven, Belgium. June 2017.

Software and related material

[1] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, L. De Lathauwer. *Tensorlab 3.0*. Mar. 2016. URL: `http://www.tensorlab.net`.

[2] O. Debals, F. Van Eeghem, N. Vervliet, L. De Lathauwer. “Tensorlab Demos - Release 3.0”. Technical Report 16–68, ESAT–STADIUS, KU Leuven, Belgium. Mar. 2016.

[3] N. Vervliet, O. Debals, L. De Lathauwer. *Nieuwste versie tensorlab vereenvoudigt big data analyse*. May 17, 2016. URL: http://www.kuleuven-kulak.be/nl/nieuws/nieuwste-versie-tensorlab-vereenvoudigt-2018big-data2019-analyse.

Conference abstracts

The doctoral candidate is mentioned in a total of 27 conference abstracts, of which he authored (and presented) 10 abstracts and co-authored 17 abstracts. Only those abstracts with the doctoral candidate as first author are listed.

[1] O. Debals, M. Boussé, N. Vervliet, L. De Lathauwer. “Numerical optimization algorithms for tensor-based face recognition”. Presented at the Computer Vision and Pattern Recognition conference (CVPR). Honolulu, Hawai‘i, USA, July 2017.

[2] O. Debals, M. Van Barel, L. De Lathauwer. “Using polynomials for nonnegative matrix factorization”. Presented at the 20th Conference of the International Linear Algebra Society (ILAS). Leuven, Belgium, July 2016.

[3] O. Debals, N. Vervliet, M. Boussé, F. Van Eeghem, L. De Lathauwer. “Tensorlab: A toolbox for (multilinear) data analysis”. Presented at the Conference of Deutschen Arbeitsgemeinschaft Statistik (DAGSTAT). Göttingen, Germany, Mar. 2016.

[4] O. Debals, L. De Lathauwer. “The concept of tensorization : applications in blind signal separation”. Presented at the Workshop on Tensor Decompositions and Applications (TDA). Leuven, Belgium, Jan. 2016.

[5] O. Debals, N. Vervliet, L. De Lathauwer. “Data analysis using Tensorlab 3.0”. Presented at the Haussdorf School on low-rank Tensor Techniques in Numerical Analysis and Optimization. Bonn, Germany, Apr. 2016.

[6] O. Debals, L. De Lathauwer. “Tensorization: what – how – why?” Presented at the KULAK Research Day. Kortrijk, Belgium, Nov. 2015.

[7] O. Debals, L. De Lathauwer. “About tensorization for blind signal separation: concept and illustrations”. Presented at the Workshop on Threeway methods in Chemistry and Psychology (TRICAP). Belluno, Italy, June 2015.

[8] O. Debals, L. De Lathauwer. "A survey on stochastic and deterministic tensorization for blind signal separation". Presented at 34th Benelux Meeting on Systems and Control (BMSC). Lommel, Belgium, Mar. 2015.

[9] O. Debals, N. Vervliet, L. Sorber, M. Van Barel, L. De Lathauwer. "Structured data fusion using Tensorlab: a demonstration". Presented at the International Workshop on Technical Computing for Machine Learning and Mathematical Engineering (TCMM). Leuven, Belgium, Sept. 2014.

[10] O. Debals, L. De Lathauwer. "Hankel-based signal separation of exponential polynomials". Presented at the 2nd International Conference on Engineering and Computational Mathematics (ECM). Hong Kong, Dec. 2013.

Reports

[1] O. Debals, L. De Lathauwer, M. Van Barel. "About higher-order Löwner tensors". Technical Report 17–98, ESAT–STADIUS, KU Leuven, Belgium. Apr. 2017.



FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
STADIUS — CENTER FOR DYNAMICAL SYSTEMS, SIGNAL PROCESSING AND DATA ANALYTICS
Kasteelpark Arenberg 10 box 2440
B-3001 Leuven
otto.debals@kuleuven.be
http://www.debals.com